

UNIVERSITY OF VAASA

FACULTY OF TECHNOLOGY

TELECOMMUNICATION ENGINEERING

Vasileios Kalos

DESIGNING A WIRELESS COUNTER FOR BALE PRODUCTION

Master's thesis for the degree of Master in Telecommunication Engineering submitted for inspection, Vaasa, 06 March, 2014.

Supervisor

Professor Mohammed Salem Elmusrati

Instructor

M. Sc. (Tech) Caner Çuhac

ACKNOWLEDGEMENT

I would like to gratitude my instructor of this thesis Caner Çuhac because without his guidance and his knowledge I could not have finished it. Also I would like to thank everyone that inspired me and helped me with his small attributes to that project. Finally I would like to thank our lab engineer Veli-Matti Eskonen for providing me a workplace with all the necessary equipment.

TABLE OF CONTENTS	page
ACKNOWLEDGEMENT	2
ABBREVIATIONS	6
1. INTRODUCTION	9
2. THEORY AND BACKGROUND INFORMATION	11
2.1. Serial Peripheral Interface	11
2.1.1. SPI Signal Lines for Data and Control	12
2.1.2. Importance of the SPI Connectivity	13
2.2. Available RF Modulations for eZ430-RF2500	14
2.2.1. Amplitude Shift Keying	15
2.2.2. Frequency Shift Keying.....	15
2.2.3. Phase Shift Keying	16
2.2.4. Minimum Shift Keying	16
2.3. Uploading the Data.....	17
2.3.1. Qt Creator	18
2.3.2. Upload Data.....	18
2.3.3. Upload Data to the Server	19
2.4. Magnetic Link	20
2.5. Magnetic Sensor	24
2.6. Apache Server	26

2.6.1. Forms and Rules	26
2.6.2. Installing Apache Server	27
2.7. USB Port Reading	27
2.8. PHP Code	29
2.9. SQLite3 Database	30
2.10. Generalizations	31
3. WIRELESS SENSOR NODES AND BALE PRODUCTION	33
3.1. General Description.....	33
3.2. Wireless Sensor Node.....	34
3.3. Bale Production	35
3.4. Sensor nodes in Bale Production	36
4. HARDWARE	38
4.1. Automation Overview	39
4.2. Display Data	41
4.3. eZ430-RF2500 Development Toolkit	43
4.4. Control of the Display	44
4.5. Modification Circuit	45
4.6. Device Configuration	46
4.7. Troubleshooting.....	48

5. SOFTWARE.....	50
5.1. Meanings and Functions.....	50
5.2. Initialization Process	51
5.3. Interrupts and Counting Process.....	52
5.4. Database Connectivity.....	54
5.5. Create SQLite3 Database on Windows	57
6. EXPERIMENTS AND RESULTS	58
6.1. Current Consumption	58
6.2. Validation Tests for the Reliability	58
6.3. Validation Tests for the Delay.....	59
7. CONCLUSION AND FUTURE WORK	61
REFERENCES	62
APPENDICES	65
APPENDIX 1. Initialization funtions for UART.	65
APPENDIX 2. CC2500 preferences modes.	66
APPENDIX 3.Pinout board.....	67
APPENDIX 4. SPI connection between MCU and LCD	68
APPENDIX 5. PHP code for updating data and create a simple web page.	69
APPENDIX 6. General characteristics of CC2500.	70
APPENDIX 7. Magnetic sensor and battery board footprints.	71

ABBREVIATIONS

ACLK	Auxiliary Clock
ASK	Amplitude Shift Keying
BFSK	Binary Frequency Shift Keying
BPSK	Binary Phase Shift Keying
CCS	Code Composer Studio
CPHA	Clock Phase
CPU	Central Processing Unit
DDRAM	Double Data Rate Synchronous Dynamic Access Memory
EMP	Electromagnetic Pulse
EMF	Electromagnetic Field
FSK	Frequency Shift Keying
GND	Ground
IF	Intermediate Frequency
LCD	Liquid Crystal Display
LED	Light-Emitting Diode
LNA	Low Noise Amplifier
LPM	Low Power Mode
MCLK	Master Clock
MCU	Micro Controller Unit
MFSK	Multilevel Frequency Shift Keying

MSK	Minimum Shift Keying
PC	Personal Computer
PCB	Printed Circuit Board
PSK	Phase Shift Keying
PWM	Pulse Width Modulation
RF	Radio Frequency
Rx	Reception
SNR	Signal to Noise Ratio
SPI	Serial Peripheral Interface
TI	Texas Instruments
Tx	Transmission
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
V	Volts
VCC	Voltage Control Collector
WSN	Wireless Sensor Network

UNIVERSITY OF VAASA
Faculty of Technology

Author:	Vasileios Kalos
Topic of the Thesis:	Designing a Wireless Counter for Bale Production
Supervisor:	Mohammed S. Elmusrati
Instructor:	Caner Çuhac
Degree:	Master of Science in Technology
Degree Programme:	Degree Programme in Information Technology
Major of Subject:	Telecommunication Engineering
Year of Entering the University:	2012
Year of Completing the Thesis:	2014

Pages: 71

ABSTRACT:

The immerge rise of crop production force engineers to design efficient ways for checking and inspecting the agricultural machines. Smart sensors that can monitor the temperature, operating time or even count the amount of the produced bales can fulfill these requirements. This can be realized by installing a monitoring device for the driver in the cabinet of the tractor that communicates over a wireless channel with the smart sensors mounted on the agricultural machine.

The target of this thesis was to develop the software and hardware of a device that is able to monitor the bale production process with the help of wireless sensor nodes to enable an easy installation process and low installation costs. The device should also be able to log and save the amount of the produced bales in a database which can be accessed from a certain website. This leads to the advantage that the crop bale production can be observed from almost every place in the world. The thesis describes the implementation of the hardware and software for a bale monitoring device and shows the results of the power consumption of a wireless node. Furthermore the reliability of the wireless channel between two nodes has been investigated.

KEYWORDS: Wireless Node, Crop Production, Bale Counter

1. INTRODUCTION

The automation of bale production machines has become a revolutionary way of monitoring the production line in the field of agriculture. Electronic chips with electromagnetic signals constantly check the production so any single mistake can be avoided. These systems are being improved day by day and there is a huge variety in market but still their production is on an increasing rate. These sensors can be established easily upon vehicles and tractors and monitor whatever conditions we choose to inspect within the field. Wireless sensor monitoring is now at an alarming rate of use with different purposes in every part of the world.

In every single field where technology occurs, there is a sensor that monitors different parameters (counting the number of bales, checking the moisture in the field etc.) of a production process. It is a good way to know how under certain circumstances the devices perform. A wireless node is a device consists of two parts: a transmitter which sends information and a receiver. Installing one of these nodes in a bale machine we can count the production process (Green 1995).

By upgrading our node, we can monitor the production in bale machines. This can be done by installing a device in the cabinet of the tractor, which displays the data received from the wireless sensor node mounted to the agricultural machine. The counting process for the bale production can be realized with a magnetic sensor attached to wireless node which is mounted on the agricultural machine.

The purpose of this thesis is to design, program and implement a system with wireless sensor nodes in such way that the crop productivity in a field can be monitored every time. This makes the procedure of the inspection easier without fault measurements. The used wireless node eZ430-RF2500 makes the work easier as it is developed for students and it still can be used for professional applications. The End device (transmitter part) will be triggered by the magnetic sensor when a movement is detected and it will send through RF communication the pulse to the receiver part. After the

wireless transmission the value of the counter will be displayed on the LCD (Stallings 2007: 78).

Moreover, the inspector of the field production can monitor the process of the crop line via Internet whenever the data are uploaded to a server. This can be done by creating a database, which contains date, time and the counter value, and webpage which reads and displays the data from the database.

The thesis is structured as follows: Chapter 2 contains the information about the mechanisms used to integrate the project. In chapter 3, the wireless communication principles are introduced. The hardware of the system is described in chapter 4, and the software development is presented in chapter 5. In chapter 6 the experimental results are shown. The conclusions are given in chapter 7.

2. THEORY AND BACKGROUND INFORMATION

The device which is used in this project is a complete development tool. It offers great potential in the field of wireless communication. It is supported by software available from the Texas Instruments web page, available with examples, included libraries, for every purpose of variable monitoring. The combination of the hardware and the software gives the user the ability to establish a wireless communication with two devices. The ez430 RF2500 tool is comprised of an End device working with batteries and a USB plugged device which is the receiver (see **Figure 1**).

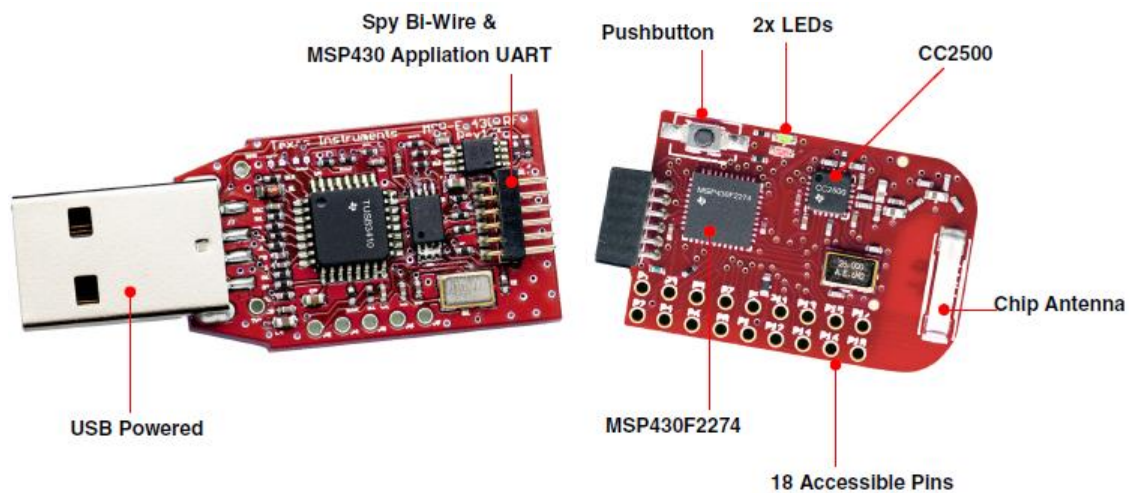


Figure 1. Wireless node general description (Texas Instruments SLAU227A 2007).

2.1. Serial Peripheral Interface

One of the most important functionalities this wireless node has is the SPI mode. Serial peripheral interface is part of our microcontroller that the node uses and makes us generate communication with the outside world so we can see the results. That mode

simply generates a relationship between two devices using a master device and a slave. In our case we have only one slave but with this mode we can use more than one slave. This method is our guide to connect two devices and make them communicate and send data two both directions (Full Duplicity Signal).

Four lines are being used in order to create the connections between the End device which is the transmitter and the peripheral (slave) device. These cables have different functionality and each one of them awakes specific duties for each microcontroller.

That type of connectivity neither diminishes the performance of the device nor consumes more current. Measurements of the current consumption will be detailed shown on forthcoming chapters.

2.1.1. SPI Signal Lines for Data and Control

This is the connectivity between the master device to the peripheral and slave device. According to that approach the modified circuit becomes a vital new expanded part.

Figure 2 illustrates the connection between Master and Slave device.

1. Master In Slave OUT (MISO) – Data output from slave device to master device.
2. Master Out Slave In (MOSI) – Data output from master device to slave device.
3. Serial Clock (SCLK) – clock signal for synchronization data.
4. Slave Select – The line to choose what peripheral device to use (the signal is selected to the Chip Select pin from the slave device).

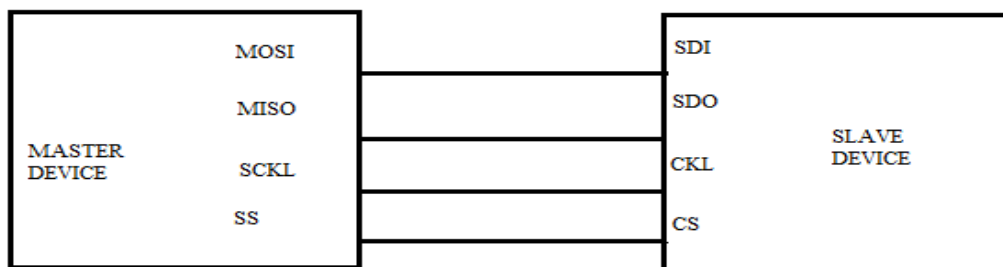


Figure 2. SPI connectivity.

2.1.2. Importance of the SPI Connectivity

These are the specific lines which must be connected in order to create the SPI mode. That functionality offers to the user easily establishment of the program through the Code Composer Studio from Texas Instruments company. Moreover, the code allows the lowest energy consumption so our circuit will not generate much current consumption. These are the main reasons to follow the architecture of the Serial Peripheral Interface.

The master device, in the wireless case, which is the USB part through SPI connectivity, enables the initialization of the functionalities of the MCU (both for Radiofrequency microchip and microprocessor chip). Also, defines the conditions of the Clock which is used for the SPI Mode. In that case, the clock it refers only for one slave device.

Advantages of SPI Mode:

- Easy to implement
- Same clock use from the Master device
- Both sides communication (MOSI- MISO)

Disadvantages of SPI Mode:

- Limited ports for different functionalities
- Short distance bus lines

That part will be expanded with the sensor. That device is our sensor transmitter which wakes up when detects a bale produced from the tractor machine. When the interrupt comes, sensor activates and transmits a signal. That signal is first saved in the Radiofrequency microcontroller. After that part, data comes to the microcontroller of the End Device and through the SPI Mode inserts the value to the LCD. Via Code Composer Studio, this value can be read as integer value 0, 1, 2, 3....as a counting procedure. With that way, data are in a readable form from the tractor user.

Figure 3 shows the receiver part which is going to be placed in the back side of the tractor.

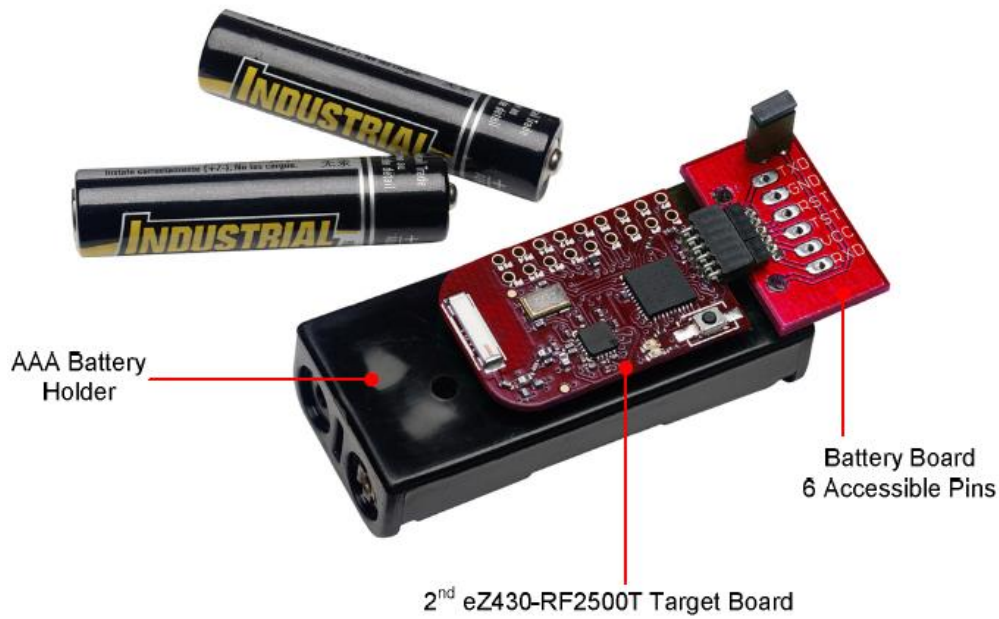


Figure 3. End device (Texas Instruments SLAU227A 2007).

2.2. Available RF Modulations for eZ430-RF2500

There are several types of radiofrequency modulations that our microcontroller uses. According to the circumstances that the magnetic sensor will create to the MCU (Micro Controller Unit), the device chooses the most suitable modulation technique to transfer data without loss (Lathi 1998: 595).

Three basic forms of modulations take part in the transfer of data in order to produce the connectivity between transmitter and receiver part. Amplitude Shift Keying (ASK), Frequency Shift Keying (FSK) and Phase Shift Keying (PSK) (Stallings 2007: 151–161). These are the basic types which eZ430-RF2500 uses for the wireless communication.

2.2.1. Amplitude Shift Keying

In that type of modulation technique the microprocessor uses two binary values in order to represent the same signal. This is a simple way to make the transmission of the signal successful as when the amplitude is zero there is no signal and when it is one then that form is the one that will be used to represent the data from the receiver's part. Still that technique is not recommended for wireless communications as abrupt changes in amplitude makes the flow to change and the original signal to lose its power.

$$x(t) = \begin{cases} A\cos(2\pi f_c t) & \text{binary 1} \\ 0 & \text{binary 0} \end{cases} \quad (1)$$

2.2.2. Frequency Shift Keying

The same logic makes the FSK modulation able to generate communication path for the transmitted data. Two different frequency values around the carrier's frequency follow the original signal to represent the binary values (0, 1). After the transmission the receiver's microcontroller chooses the most suitable frequency to convert the data from digital to analog form (Lathi 1998: 598).

$$x(t) = \begin{cases} A\cos(2\pi f_1 t) & \text{binary 1} \\ A\cos(2\pi f_2 t) & \text{binary 0} \end{cases} \quad (2)$$

The instantaneous frequency corresponded to the Fourier series can be related to:

$$\omega_i(t) = \frac{d\theta}{d\tau} \quad (3)$$

and

$$\theta(t) = \int_{-\infty}^t \omega(\alpha) d\alpha \quad (4)$$

2.2.3. Phase Shift Keying

In this technique the carrier signal is shifted in phase according to the transmission rate. The simple form of that modulation generates two different phases for two binary values. This is the protection mechanism to make sure that every single data can reach its final destination. If that cannot be done, the value of zero will be erased and the signal with value one will take place.

$$x(t) = \begin{cases} A\cos(2\pi f_1 t) \\ A\cos(2\pi f_2 t + \pi) \end{cases} = \begin{cases} A\cos(2\pi f_c t) \\ -A\cos(2\pi f_c t) \end{cases} \begin{matrix} \text{binary 1} \\ \text{binary 0} \end{matrix} \quad (5)$$

2.2.4. Minimum Shift Keying

The specific type of modulation allows the RF communication chip to use efficiently the spectrum of bandwidth, making the dispatch of data more accurate and stable. There is a small percentage of error for MSK that is why often it is mentioned as Multilevel Frequency Shift Keying (MFSK). The Tx signal is given from the equation:

$$x(t) = \begin{cases} \sqrt{\frac{E_b}{T_b}} \cos(2\pi f_1 t + \theta(0)) & \text{binary 1} \\ \sqrt{\frac{E_b}{T_b}} \cos(2\pi f_2 t + \theta(0)) & \text{binary 0} \end{cases} \quad (6)$$

With E_b is given the transmitted energy for each bit when T_b is the bit duration, $\theta(0)$ is the phase at $t=0$. To simplify the meaning of that type of modulation that the RF microcontroller uses, it is a continuous detector of signal to reassure that the data which came “that” moment is the correct one.

The detection period which satisfies the MSK is the following:

$$f_1 = f_c + \frac{1}{4T_b} \quad (7)$$

$$f_2 = f_c - \frac{1}{4T_b} \quad (8)$$

and the carrier's signal is given from the mathematical type:

$$x(t) = I(t)\cos\left(\frac{\pi(t)}{2T_b}\right)\cos 2\pi f_c t + Q(t - T_b)\sin\left(\frac{\pi(t)}{2T_b}\right)\sin 2\pi f_c t \quad (9)$$

as a multiplied signal with the carrier wave.

2.3. Uploading the Data

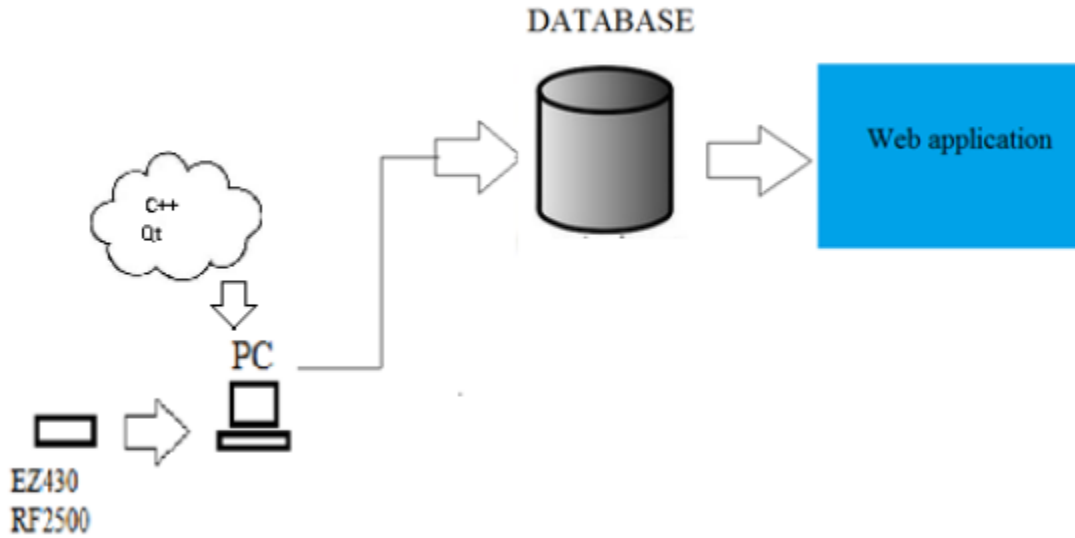


Figure 4. Connectivity between different programs.

The immerge rise of monitoring the production line enables daily engineering to upload data to a server as shown in **Figure 4**. That makes the head manager of the production line to check the results any time. Creating a database with date, time and the current value of the production unit we can generate a Web page which displays the necessary values, making the monitoring of the bale production visible from any part of the world. The initialization of this procedure will be examined on the next chapters.

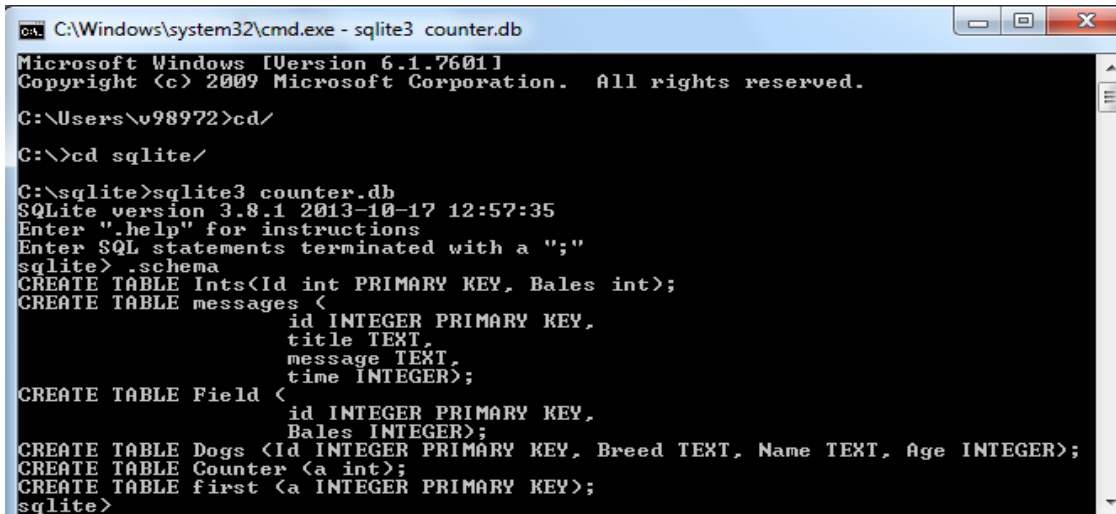
2.3.1. Qt Creator

There are plenty of methods to initialize a database and use it to update the desired data. The most efficient, easy and fast way to response in such challenge was through Qt Creator and Sqlite type of database. Easy functions inside Qt Creator, makes the software easier. Ready functions allow a user to connect instantly with the database. Using the specific commands desired information updated the specific moment without loss. Test and measurements will reassure the results.

An effective way to create a database is the SQLite3 software through Qt Creator. The most common way is to set up a .db (database type) with Command Prompt via Windows and then include that path in the main program of Qt Creator. Second, Xampp should be installed so that we can upload our database to the PHP Server and create a table.

2.3.2. Upload Data

The creation of the database can be done easily through Command Prompt on Windows using a soft version of SQL libraries, the SQLite3. First important detail is to download and put inside C:/ the file which contains the sqlite3 creator .exe file. After that step the user can open CMD from Windows environment and open the program. This is a program to create the specific type of database which is going to be inserted into the Qt functions. **Figure 5** represents the SQL commands used to create a table in a database.



```

C:\Windows\system32\cmd.exe - sqlite3 counter.db
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\v98972>cd/

C:\>cd sqlite/

C:\sqlite>sqlite3 counter.db
SQLite version 3.8.1 2013-10-17 12:57:35
Enter ".help" for instructions
Enter SQL statements terminated with a ";"
sqlite>.schema
CREATE TABLE Ints(Id int PRIMARY KEY, Bales int);
CREATE TABLE messages <
    id INTEGER PRIMARY KEY,
    title TEXT,
    message TEXT,
    time INTEGER>;
CREATE TABLE Field <
    id INTEGER PRIMARY KEY,
    Bales INTEGER>;
CREATE TABLE Dogs <Id INTEGER PRIMARY KEY, Breed TEXT, Name TEXT, Age INTEGER>;
CREATE TABLE Counter <a int>;
CREATE TABLE first <a INTEGER PRIMARY KEY>;
sqlite>

```

Figure 5. Sqlite3 database command lines for creating a table.

This is the command line which follows to open the desired database, in that case the counter.db (this is the type of library which can be inserted into Qt and works properly) and its table contents is Counter (an int). This is the value which will be updated automatically every time Qt runs and opens the Serial Communication with the Receiver (part which takes the data wirelessly and deploy them to the LCD). Every time Qt awakes that library, contents have access through PHP connector. A small diagram shows the philosophy behind that procedure.

The database is created with the desired types of contents inside it (integers which increment by adding a new row, strings with a message etc.). The most important step in uploading the constructed database is the path of it which should be placed into Qt Creator so that can always be in a “read” state from the main program.

2.3.3. Upload Data to the Server

Introducing Qt Creator, it is innovative software which helps users to create a useful and easy interface to deal with an embedded device. A variety of functions and libraries enables users to interfere with different levels of communication via the same program. For example, as the following chapters will explain, Qt Creator can access the USB Serial Port and read the incoming data. Later, these data can be added to the database

and constantly updated. Through Xampp which is an apache server application, creation of a web page and insertion of these data is the last stage of that project. All these programs above should be synchronized so that the desired result will appear to the web page. In next chapters detailed procedure will be explained for every single step.

Qt Creator uses the same interface to handle different events the same time without interfering between them. In addition, within the same program data could be uploaded to the web page directly from the source. In our case, the source is the LCD Counter which deploys how many bales came from the tractor. Functionalities which interlink the manual database with the reading procedure of the USART information (in that case is the 16bit number) allowing data to be constantly changed. The incoming data are uploaded through the Apache Server and the creation of a simple web page is done with a note pad file and PHP code.

2.4. Magnetic Link

As the implementation of the project will be installed on a tractor, nodes should be able to communicate each other without interference, noise problems, broken connection etc. This could be done with the use of a magnet to replace the push button. The right angle pins that are going to be used are presented in **Figure 6**.

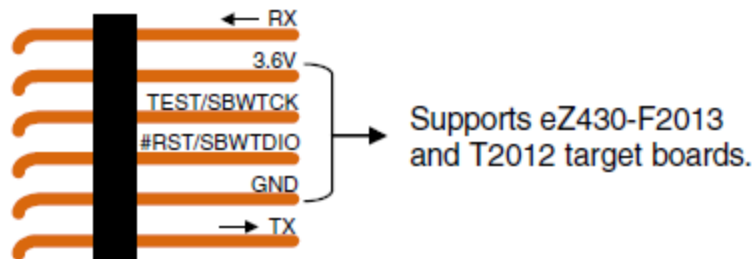


Figure 6. Battery board pin out for plugging the magnetic sensor (Texas Instruments SLAU227A 2007).

The expansion of the End Device will be done with a magnetic sensor. A useful adaptation generates an industrial automation which provides accuracy and stability. From the material of the construction, a magnet causes changes to the electromagnetic field of its surface when comes in touch with an iron, charged surface etc. This functionality is shown in **Figure 7**, will help the transmitter to send data only when reaction occurred between the magnetic sensor part and a metal object.

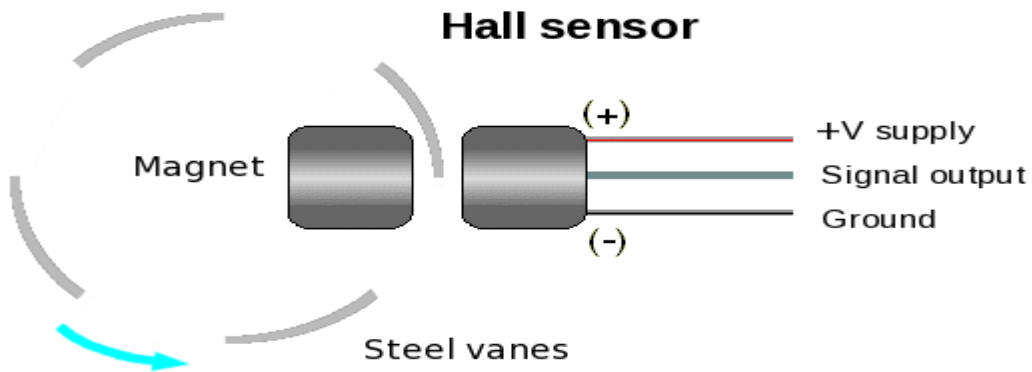


Figure 7. General description of the magnetic sensor (MegaSquirt 2013).

As the magnet comes close to the surface of the sensor electromagnetic field starts generating movement of electrodes. That relation can be viewed as a relation of Force as F and Electrical Field as E below:

$$F = -Ze E \quad (10)$$

Generation of the magnetic field will cause an increase of the electricity speed given from the relation:

$$v = \mu E = \frac{\mu}{ze} F \quad (11)$$

As the current flows from the sensor's part and the charging of the magnet surface will automatically produce the change in the particles giving the sense of the push button like being pushed from someone (Messerly 1994: 21-24).

The density of an electron is given from the relation:

$$j = n_0 e V_c = \frac{ne^2}{m_e} z E \quad (12)$$

The relationships between the electrode mass and its speed are strictly constrained because of the metal surface which creates electromagnetic field (Messerly 1994: 36-37).

Thus the conductivity which produced is given from the following equation:

$$\sigma_e = \frac{ne^2}{m_e} \tau = \frac{ne^2}{m_e n Q c_e} \quad (13)$$

Reading the data sheets from the sensor, a user could realize that every pin (magnetic sensor uses 3 legs) corresponds to a different functionality. Given that rule, the button from its side is installed and connected to the End board with a specific way, being an important factor of the whole End Device.

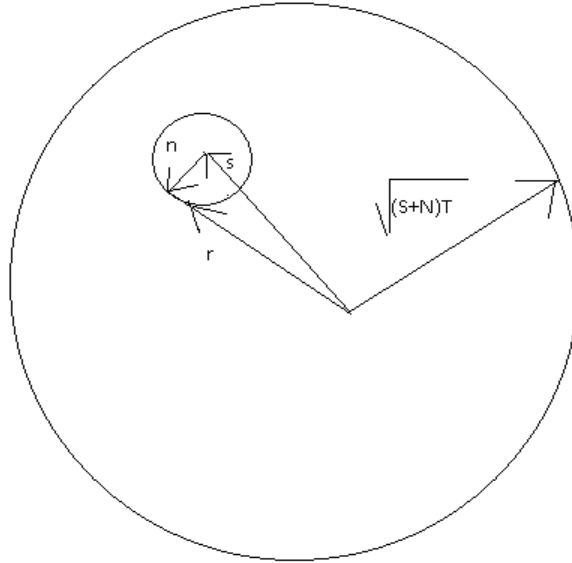


Figure 8. Example of magnetic vectors in open space.

Connecting the pin-legs of the button with the pin-legs of the sensor, the device now triggers externally when the magnetic sensor touch a metal surface. The interrupt which

created, under no circumstances should be enabled to generate the radiofrequency pulse which corresponds to increment our counter.

When an electromagnetic field (EMF) generated in free space it can be depicted as a vector with velocity, trend etc. Vector x' is the electromagnetic field which will modify the polarity of the sensor. That vector is represented in the **Figure 8** (Lathi 2008: 713).

Figure 9 represents the generated electromagnetic field with the help of vectors.

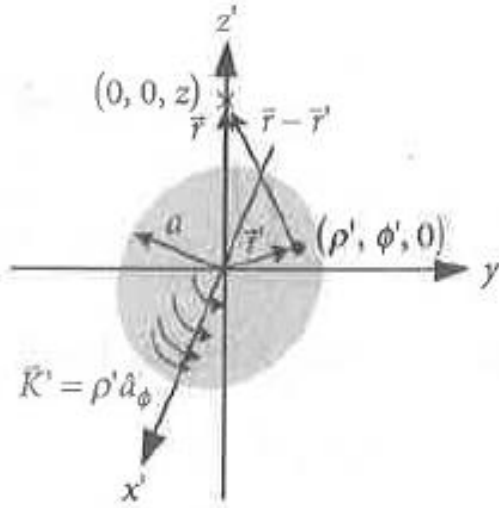


Figure 9. Generated electromagnetic field with vectors (Kaiser 2004: 23-22).

The magnetic field is represented as the x' vector along the z axis. Having this general description of the electromagnetic energy, a mathematical approach can be evaluated from the Cartesian coordinators as the energy released in free space related to time:

$$\overline{H}(0,0,z) = \frac{1}{4\pi} \widehat{\alpha_z} \int_0^{2\pi} \int_0^\alpha \frac{(\rho')^3 d\rho' d\phi'}{[(\rho')^2 + z^2]^{\frac{3}{2}}} + \frac{z}{4\pi} \int_0^{2\pi} \int_0^\alpha \frac{(\rho')^2 d\rho' d\phi' (\cos\phi' \widehat{\alpha_x} + \sin\phi' \widehat{\alpha_y})}{[(\rho')^2 + z^2]^{\frac{3}{2}}} \quad (14)$$

As a general form exists that the magnetic vector of a free space generated field can be given after simplify the equation above as:

$$\left[\frac{(a^2 + 2z^2) - (2z\sqrt{a^2 + z^2})}{(2\sqrt{a^2 + z^2})} \right] \hat{a}_z \quad (15)$$

because $\hat{\alpha}_x$ $\hat{\alpha}_y$ remained zero from the previous equation. This is the EMF in free space.

2.5. Magnetic Sensor

The sensor which is going to be used to complete the expansion of the End Device has the form as shown in **Figure 10**.

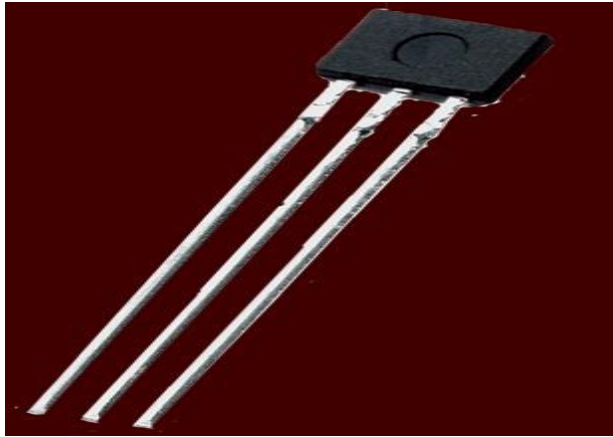


Figure 10. Magnetic sensor for substituting the push button (Mouser Electronics 2014).

That type of sensor can easily be connected with the End Device because it has 3 pin-legs operated at low voltage of 3V (the Node uses 3.3V). The polarity changes as the magnet comes in touch with the surface of the sensor as described in **Figure 11**.

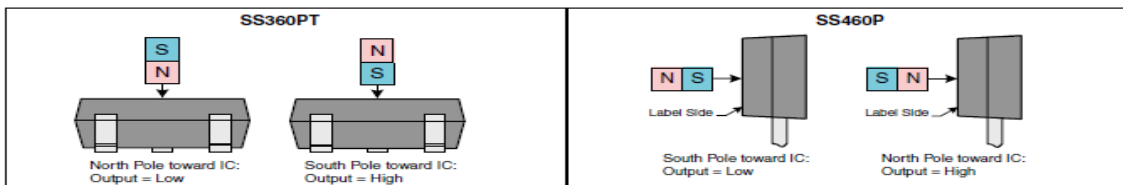


Figure 11. Poles of the magnetic sensor (Honeywell SS360PT/SS460P 2013).

Magnetic sensors detect changes and disturbances in a magnetic field like flux, strength and direction. Other types of detection sensors work with characteristics like temperature, pressure, light. From established knowledge about the existing magnetic field and the data collected from sensors regarding changes and alterations, many things can be known. Rotation, angles, direction, presence and electrical current can all be monitored. Magnetic sensors are divided into two groups, those that measure the complete magnetic field and those that measure vector components of the field. The vector components are the individual points of the magnetic field. The techniques used to create these sensors involve various combinations of physics and electronics (Stallings 1994).

The magnetic vector potential which is produced from an electromagnetic field upon a metal surface can be given from a quite perplexed form as (Kaiser 2004: 23):

$$\vec{A}(r) = \int_{L'} \frac{\mu_0 \vec{I} dL'}{4\pi(\vec{r}-\vec{r}')} + \iint_{S'} \frac{\mu_0 \vec{K} dS'}{4\pi(\vec{r}-\vec{r}')} + \iiint_{V'} \frac{\mu_0 \vec{J} dV'}{4\pi(\vec{r}-\vec{r}')} \quad (16)$$

Still there can be detected a sensitivity in the contact without generate the right impact to transfer the electromagnetic pulse (EMP) because of the high temperature generated from electrode collisions. This is called energy loss mechanism and can be expressed through the following type:

$$W_c = \frac{3}{2} \delta k(T_e - T_g) \quad (17)$$

corresponds to collision loss between electrons with T_e and T_g the temperatures of ionization of an electron . Since δ is the average energy loss per collision which is given from the type:

$$\delta \approx \delta' \left(\frac{2m_c}{m_g} \right) \quad (18)$$

where m_c is the electron mass and m_g is the heavy particle mass.

2.6. Apache Server

The tool for initializing a Web server and create the communication links between the PHP code and the Qt Creator which runs the database is the Apache server. Nowadays more than 53% of web services run through the XAMPP application. Apache does not have a specific architecture, allowing the user to implement the necessary methods, coding blocks and infrastructures so that the creation of a Web Page could be established (Yank 2004).

2.6.1. Forms and Rules

In order to create a database, the main step is to generate a server or to establish a server on one single computer or even more than one PC (Personal Computer). Server, a “software and specific hardware” (Wikipedia), as the name says is a system which functionalities are dedicated to respond to specific orders and help the user accomplish these actions on the internet and not only. The importance behind a Server is that shares common resources to the users of that system.

Despite the generalization of the meaning, still that does not mean that is an easy procedure because the user should know exactly what will be the priorities of that server and what tasks is going to execute. For instance, in the case of that project the working environment is mainly on Windows 7 which means that all the programs which should run must be compatible to Windows and only so that there will be total communication between the various functionalities.

The installation of a Server and uploading database on a Linux operating system may be approachable and easy for an embedded software user but it is a challenging part to make always an advent way and explore new methods.

2.6.2. Installing Apache Server

The first step that should be done is to download the Xampp for Windows which is free of charge and easy to find it from the Internet. Using an Apache Server makes coding reliable and easy to implement application which uses the infrastructure (MySQL, PHP etc.).

Figure 12 displays the libraries needed to establish a communication between SQLite and the webpage that displays the data.

pdo_mysql	
PDO Driver for MySQL	enabled
Client API version	mysqlnd 5.0.11-dev - 20120503 - \$Id: 40933630edef551dfaca71298a83fad8d03d62d4 \$

pdo_sqlite	
PDO Driver for SQLite 3.x	enabled
SQLite Library	3.7.7.1

Figure 12. Library types inside PHP and Apache server.

MySQL type of database needs to be uploaded and used for the Qt Creator application. This application is used to read the data from the USB Port of the computer (Receiver Part is connected on USB Port) and constantly renew the value of them so that the supervisor of the production line of the field can access them and read how many bales produced a specific day. Detailed explanation will be given to the forthcoming chapters regarding the software (Meloni 2012: 357).

2.7. USB Port Reading

Daily technology gives many advantages to a user that desires to generate an effective product which can easily launched to markets. For that reason, the project should give

the potential to handle data (counter value, date and time) and make them reachable to anyone who needs to watch these data. Qt Creator, with its flexible interface allows a user to construct a mechanism which reads from a USB Port which is in use on a computer what data travel. These data can easily store as a general value which does not change its type while the main program is executed and stored in such way that the database can access them. **Figure 13** represents the Rx part which is going to be placed on the cabin of the tractor. It should be mentioned that, the eZ430-RF2500 node is being read from the USB port of a computer. It is to mention that there is a USB to serial converter on the node that converts USB signals into UART signals and vice versa. The node has a USB to a serial converter microchip. With that way the device can be detected and used as a normal USB driver.

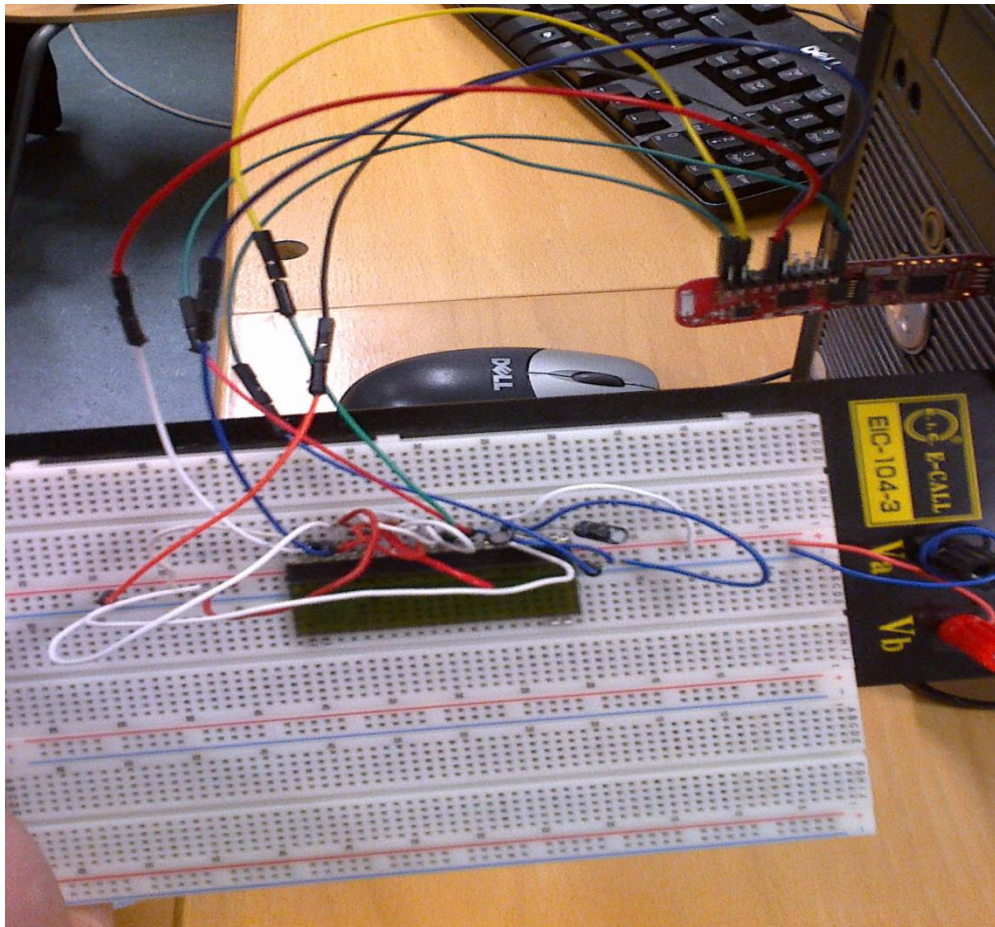


Figure 13. Receiver part of the wireless node.

Using a program called RealTerm a user is allowed to specify the Baud Rate, USB Port and form of the received data in order to read in which type (Decimal, Hexadecimal, etc.) that specific data are and convert them within the program to the desired form. This should be done for the reason that every program and device uses different types of data inside the microprocessor regarding of how much efficient these types are. That is the reason to convert them back again to Decimal inside Qt Creator, so that uploaded data will have the form that someone could understand what he reads (for instance, how many produced today is a number from 1 to 1000).

2.8. PHP Code

That kind of scripts such PHP (note pad, text forms) can accurately design and form server functionalities and give the programmer a variety of potential to insert data, to leave the uploaded page, insert images and a wide variety of other uses. PHP mainly focuses on three areas (Welling & Thomson 2008: 14).

The most challenging part is the site scripting, where the user has vast majority of choices to satisfy the web needs according to his potential. A user can access the PHP program with a web browser, viewing the PHP page through the server. In that case, an Apache server is used.

Command scripting: Programmer can make a PHP script to run it without any server or browser. You only need the PHP parser to use it this way. This way of scripting makes it a competitive way of using Task Scheduler on Windows. These scripts can also be used for simple text processing tasks.

Writing a PHP application may not be the best choice to handle a web page but because of the flexible infrastructure which is provided, a user can deploy easily data forms from third party applications. The user also has the ability to form cross-platform applications with this way.

Coding with PHP allows easy use of Xampp so that a user can marry different infrastructures. That means different types of software and combines them with a unique way without confusing data or distorting them. PHP software language is used in that case to read the database and the desired data so that the server will upload them on the Internet.

The database is SQLite3 which can easily be accessible from Qt Creator and created on the system from a user. One main step that should not be avoided is what type of data a user should generate. In the case of that project, once a SQLite3 database created, the user should first think about the types of data. To begin with, the counter value should be an integer which takes values from 1 up to how many bales produced that day. In addition, date and time is a specific type of data which can be easily inserted into the database of SQLite3. Further information will be explained in chapter 5 regarding the software.

2.9. SQLite3 Database

An effective way to access a database through Qt Creator is to generate a SQLite3 type database which is actually a simpler edition of a MYSQL database type. The user interferes with the database link as described in **Figure 14**.

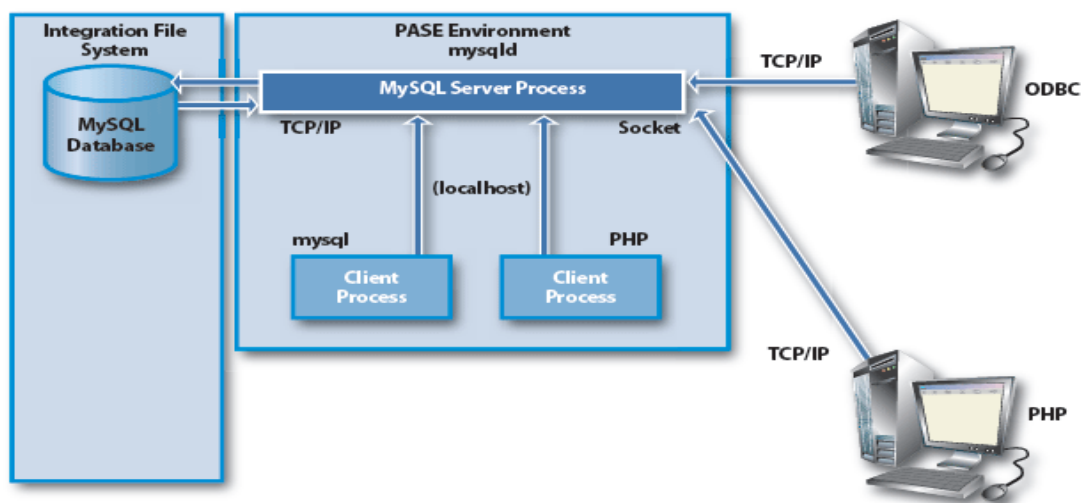


Figure 14. SQLite3 communication with PHP user interface (Prodeveloper 2014).

First step is to download the application which is free of use on the Internet and store it inside C in a file. Next, if the user will open the <cmd> he/she can manually operate with the SQLite3 database generating a base with the exact contents that the operation requires. Types of commands such <select * from “class name”> or <sqlite3 create “database name”> will be explained in the chapter 5. For instance, in the specific project the counting value should be saved in such way that this value will be always up to date. Furthermore, it would be beneficiary if the Line Inspector could see the date and time of the work so any kind of mistake could be avoided. Detailed code will be given in the chapter 5.

2.10. Generalizations

The user of such automations should be able to handle all these functionalities above meticulously and with detailed approach for the desired result. The way that should be followed must be a synchronized one in order to get the results in real time. For that reason, the cooperation of MySQL, PHP and Apache server gives the user the best combination to create an environment to update all the necessary information in such way that everything will be saved (Stucky 2002).

In addition, the operation process of the UART read and write enables the user to generate a quick and stable way to save the desired value and pass it through the update functions. A general description of a standalone web application is represented in **Figure 15**.

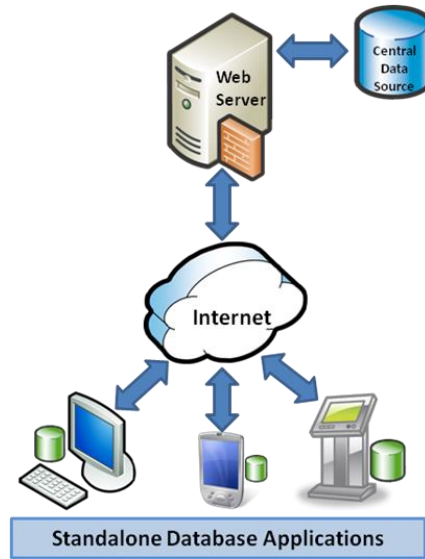


Figure 15. Cooperation of applications (Microsoft Sync Framework Developer Center 2014).

This is the logic behind this software so that everyone can access the created database from wherever can get access on Internet. Detailed approach will be given through the next chapters step by step so that the procedure will be detailed explained.

3. WIRELESS SENSOR NODES AND BALE PRODUCTION

3.1. General Description

Figure 16 shows the general form of a wireless sensor node.

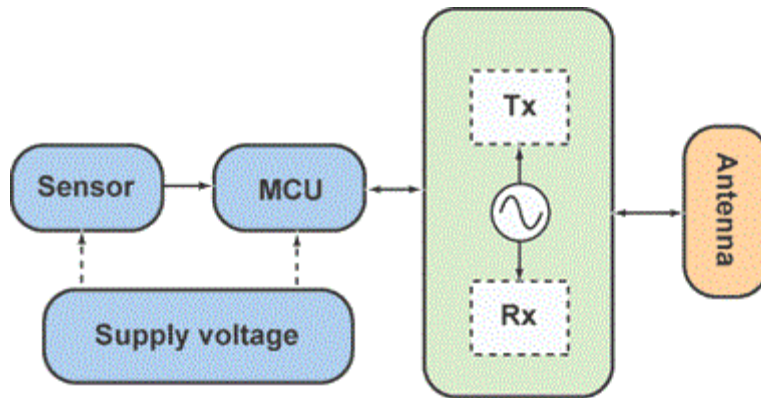


Figure 16. Architecture of wireless sensor node (Microwaves & rf 2014).

The sensor generates electromagnetic field and when the magnet side touches an iron surface, an interrupt arose the microcontroller and sends data to the receiver (Bowick 1997).

MCU (Microcontroller Unit) will manipulate the generated pulse and within SPI mode will enable the communication link sending through RF (Radiofrequency microcontroller) chip to the Receiver part.

3.2. Wireless Sensor Node

As communication is one of the most important factors to make our lives easier, innovations as wireless sensor networks take part in many areas. These devices are commonly used to send data from one part to another (McGraw 1999). The transmitter is an electronic device which under specific programming can produce data and send them directly to the receiver. This device can be also called End device and can easily connect in the USB flash drive of a computer. To the other side, we have a self-preserved electronic device with battery which collects data and it is the receiver. This is a complete wireless node.

This wireless network can be deployed with more than one receivers placed in space according to the range that the node can provide. In our case the transmission can be successful within the range of 30 meters. The connection between battery board and eZ430-RF2500 is shown in **Figure 17**.

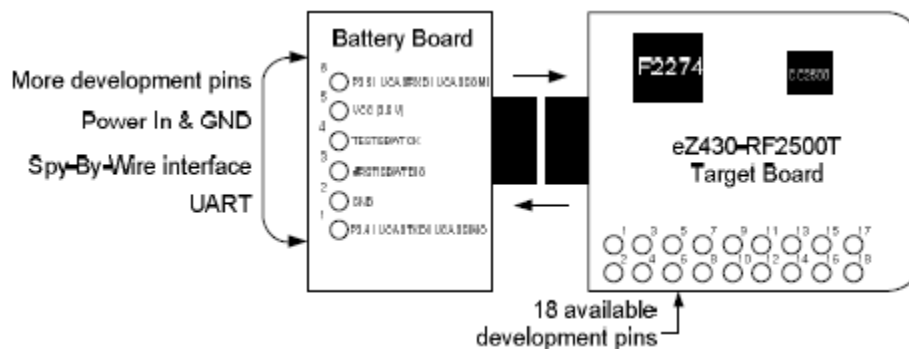


Figure 17. Connection between battery board and eZ430-RF2500 (Texas Instruments SLAU227A 2007).

3.3. Bale Production

This wireless sensor device can easily be adopted into the back side of the tractor. The transmitter's part will be installed in the back part of the bale machine where the sensor will monitor each bale that is going to come out the machine (see **Figure 18**). The second part will be installed to the driver's panel with a display so the driver can see how many bales produced during his work.



Figure 18. Bale production with tractors (Ranch Ramblins 2007).

The used wireless node has a great potential in the crop production. If specific sensors installed in the already given structure of the board, not only monitoring of weather conditions (sun, rain, fog etc.) but also the field's circumstances could be measured (humidity, mud terrain etc.) in order to protect the machines. **Figure 19** shows the parts of the tractors where the nodes will be placed.

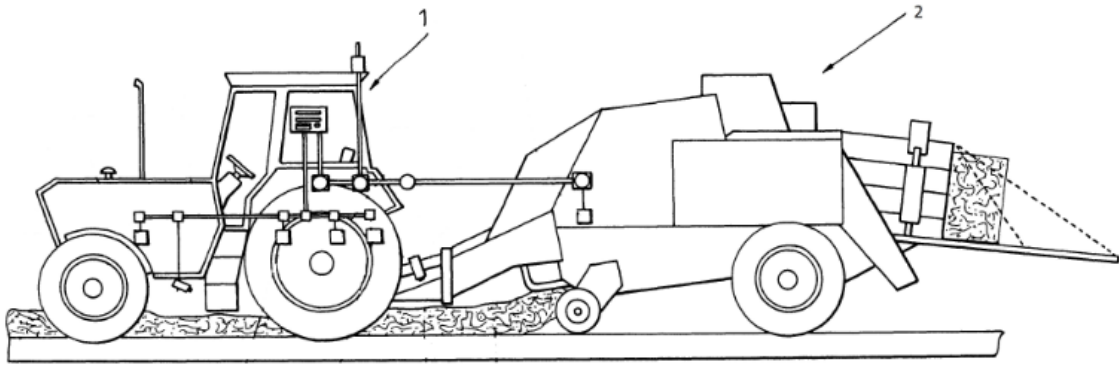


Figure 19. Implementation and installment of wireless sensors on the tractor (Google 2012).

This is a general form of how nodes can be placed in order to inspect and monitor the productivity of a bale machine. The receiving part will be placed on the driver's table. The transmitter (End Device) will be placed on the back side of the tractor part so that when a bale comes out the magnet will detect motion and send wirelessly the data.

3.4. Sensor nodes in Bale Production

Fields have vast space without obstacles to block the transmitted signals from the End device allowing the full spectrum of 2.4 MHz for use without propagation effects.

The coherence of a channel because we have only one user can be given from the mathematical type:

$$P_b = \frac{1}{2} e^{-\frac{\frac{1}{2}E_b}{N}} \quad (19)$$

This type of channel coherence we find it similarly to almost every kind of shift keying modulation because of the binary way of transmission (Lathi 1998: 599-601).

With that way the inspection of the production line becomes faultless without human interaction just by changing shifting the logic zero (when the magnetic sensor does not attach metal surface) to logic one (interrupt enabled when metal detected and sensor reacts to the current flow).

Security takes part in the installment of a wireless node as the device is completely harmless to human interaction. On the other hand, both of transmitter and receiver parts are easily implemented on the tractor due to their small size. Due to the growth of renewable resources of energy alternative ways should come to the market so that cost will be diminished and life expectancy of electronic devices will be increased. For the reason that the wireless transceiver uses the spectrum of 2.4MHz is safe to environment and the human, these forms of communication will be upgraded and replace daily technology which consumes energy and cost (Harikrishnan 2010: 8-9).

4. HARDWARE

The main reasons to use that device are the low cost equipment, the efficient code that can be implemented to open the wireless communication, the expansion of the life of that device. Last but not least, it is an effective way to develop the hardware. Using the LCD EACODM16E – A, we can display the incoming wireless data in an efficient way that only when an interrupt comes from the End device will send it. More details will be explained in the forthcoming chapters. The completed LCD circuit connected with the Rx node is presented in **Figure 20**.

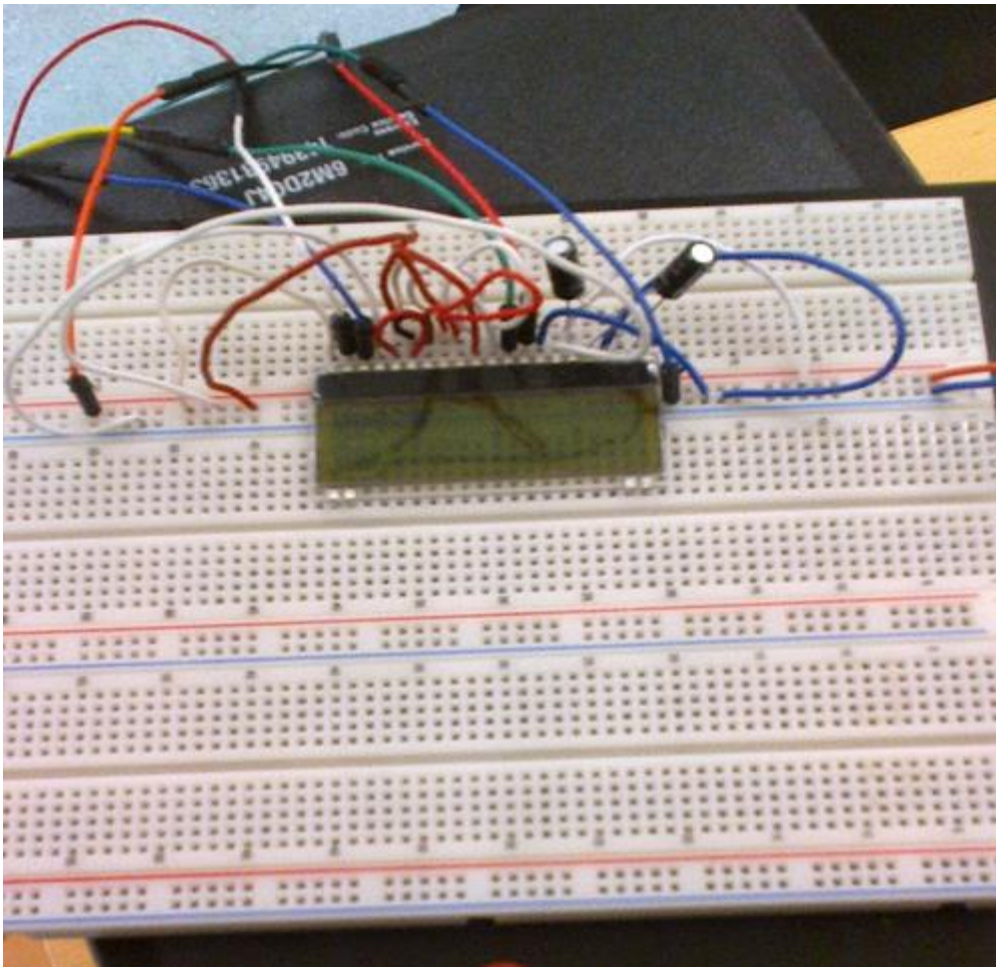


Figure 20. Circuit for LCD Display.

This is the part which is going to be installed on the driver's console so that while he is driving he can automatically see the progress on the LCD.

4.1. Automation Overview

The wireless counter applied on bale machine is a revolutionary way to make the line production work more powerful without mistakes which cost. The core of that project is the unique traits of the wireless development toolkit of eZ430-RF2500. The LCD which is used for displaying data is shown in **Figure 21**.



Figure 21. LCD used for displaying incoming data (Electronic Assembly 2012).

To begin with the transmitter part, the designer does not change anything in the electronic circuit of the wireless node, just expand it by adding the magnetic sensor which will count the bales. This could be implemented by using the pins of the transmitter's microprocessor. With that way, when the sensor changes its value from zero to one that means that a bale has been produced and then current send data to the MCU.

The data flows from the microprocessor of the device to the Tx and sends the wireless data to the Rx. If the dispatch of data is valid, the information is on the receiver's part which means that something came (packet of information received). The change of current flow enables the process in the second part of our electronic innovation. By expanding our receiver's part to a counter with a display, our project is finished. Whatever the magnetic sensor counts, data flows wirelessly to the receiver and information is being read on the display (Greenfield 1983: 355-373).

Modification should be done to the electronic circuit in order to use the right voltage that our device uses. That means that a voltage regulator is needed in order to take the tractor's voltage and converted into the necessary voltage that the wireless node uses. In our case, the battery uses a 12V battery. Wireless node final form is given in **Figure 22**.

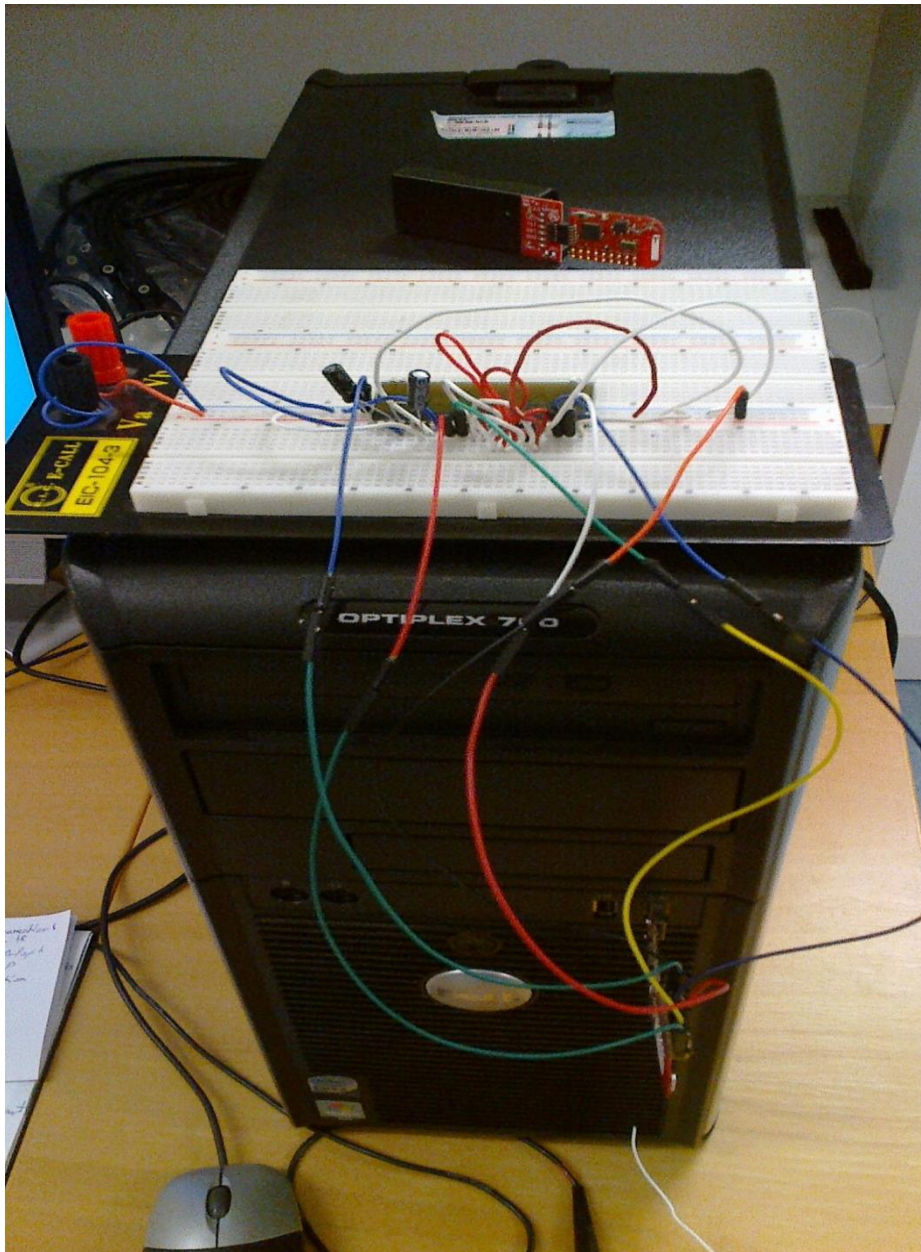


Figure 22. Receiver node upgraded with the LCD.

4.2. Display Data

Available functions are shown in **Figure 23**, allowing the user to optimize the use of the display. Using the LCD, a user should first connect it with that way that communicates through the SPI mode with the Receiver's part. Serial Peripheral Interface means, faster communication between two or more devices, less complexity, fast wireless radiofrequency transmission. This can be done by checking the data sheets of the LCD and the available communication ports of the MCU of the node.

Example of initialisation: 8 bit / 3.3V											
EA DOGM163											
Befehl	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	Hex Bemerkung
Function Set	0	0	0	0	1	1	1	0	0	1	\$39 8 bit data length, 2 lines, instruction table 1
Bias Set	0	0	0	0	0	1	0	1	0	1	\$15 BS: 1/5, 3 line LCD
Power Control	0	0	0	1	0	1	0	1	0	1	\$55 booster on, contrast C5, set C4
Follower Control	0	0	0	1	1	0	1	1	1	0	\$6E set voltage follower and gain
Contrast Set	0	0	0	1	1	1	0	0	1	0	\$72 set contrast C3, C2, C1
Function Set	0	0	0	0	1	1	1	0	0	0	\$38 switch back to instruction table 0
Display ON/OFF	0	0	0	0	0	0	1	1	1	1	\$0F display on, cursor on, cursor blink
Clear Display	0	0	0	0	0	0	0	0	0	1	\$01 delete display, cursor at home
Entry Mode Set	0	0	0	0	0	0	0	1	1	0	\$06 cursor auto-increment

Figure 23. Initialization functions of the LCD (Electronic Assembly 2012).

The connectivity should be accurate because every single fault output will have undesired results. The functions must correspond only to 3.3V and SPI mode. When the implementation circuit to connect the LCD with the Receiver drawn, with the Multisim (electronic circuits design), the designer should first create the circuit with discrete equipment (resistances, capacitors etc.). Now, we are able to communicate with an external source and deploy data.

```

<send_command_to_lcd(0x38);>
<send_command_to_lcd(0x39);>
<send_command_to_lcd(0x15);>
<send_command_to_lcd(0x78);>
<send_command_to_lcd(0x55);>
<send_command_to_lcd(0x6D);>
<send_command_to_lcd(0x38);>
<send_command_to_lcd(0x0F);>
<send_command_to_lcd(0x01);>

```

Example of how to trigger (enable bit) a specific port of the LCD to take the commands is given in **Figure 24**.

Display ON/OFF	0	0	0	0	0	0	1	D	C	B	D=1:entire display on C=1:cursor on B=1:cursor position on
-------------------	---	---	---	---	---	---	---	---	---	---	--

Figure 24. Data Array of the LCD (Sitronix 2003).

We check carefully which bits must be enabled in order to have the display on and the cursor on. If we see from that specific bite, the position C and D should become “1”. Converting the number from decimal to hexadecimal we generate our command data and send them to the display by enabling (with logic AND) the right port (0x40). With the same way all the commands to initialize the necessary functions follows to awake the LCD. The general footprint of the LCD it is presented in **Figure 25**.

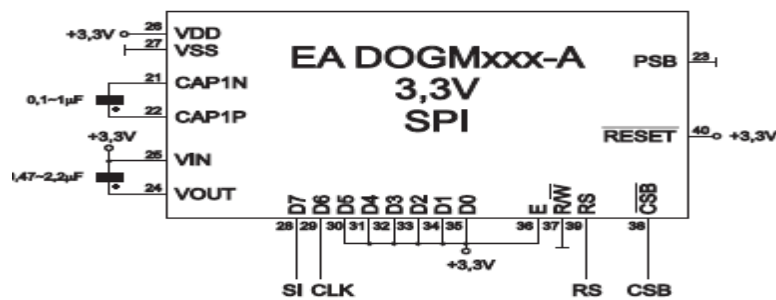


Figure 25. Pin connections of the LCD board (Electronic Assembly 2012).

It is clear from the LCD schematic above, to be initialized once, the user should carefully install all the necessary parts within the right pins of the display for proper use. Detailed connections must follow in order to keep the initialization process within the specifications for the general purpose of the automation. The user should carefully choose the circuit which corresponds to the proper voltage (in our case 3.3V) so that the connections will be compatible with the Node device (which also works with 3.3V).

With that way, the automation works properly without current loss which corresponds to malfunction of the Radio communication and other dreadful effects (destroy of materials, big current consumption, device not working properly etc.).

It may looks complicated, but with that way the user has the ultimate contribution of the LCD functionalities and the total potential of it, making it work rightfully as the user desires.

4.3. eZ430-RF2500 Development Toolkit

To the experimental part of that project, we need to adjust the receiver part of our wireless node in the USB port of a computer. The next step is to download the compatible software to our device from the web page of Texas Instruments so the device could be programmed exactly with the specific codes. When the program once installed, our device is readable from the PC and ready to compile. This is the procedure in order to give specific orders to the microprocessor unit so as it can obey to the code and act with the necessary tasks. To our case, for the transmitter part should awake the RF communication when the magnetic sensor detects motion and initialize the display and generate the counter to the receiver's part. Meticulous programming is needed in order to set up the desired functionalities for the display and generate the counting process.

4.4. Control of the Display

The display is connected with the Rx node via SPI. **Figure 26** shows how the Rx node is connected with the LCD Display.

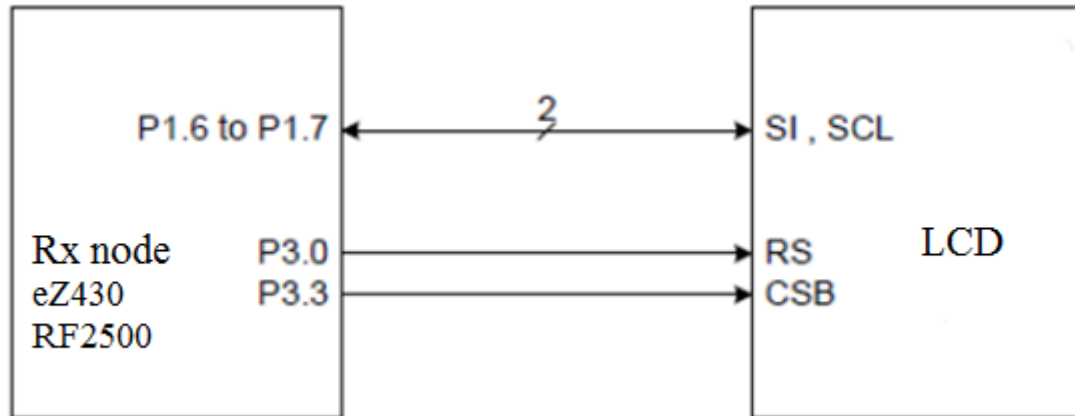


Figure 26. Pins used for SPI mode.

These functions are registry bits which activate specific traits of the LCD such as the blinking cursor, the initial position of the cursor, the open of the LCD, the DDRAM spaces we are using to allocate the numbers for the counting process, the back light of the LCD, the moving cursor etc. Reading carefully the data sheet for the display we can clearly have the image of what bits should be enabled in order to have these functions. We should activate the main port of the display so that it can understand that “now” we send some commands. For that purpose, we are using the logic AND activating the command process for the LCD. If we want to send data, then using the logic OR, deactivate that bit and push the desired data to the display (Fletcher 1980: 415-424).

For the initialization process we can create a wireless node communication by simply activate the receiver which is a device with own power supply which comes from a battery. When the receiver is ready, two Leds (Lighting Emitting Diode), one red and one green blinking constantly and gives the signal that the RF communication port is ready. Under the right code lines, the transmitter which is connected to the PC (Personal

Computer) can be programmed and give functionalities to the receiver. That is the way to set up the project for the initialized stage. The DDRAM spaces are illustrated in **Figure 27**.

1	2	3	4	5	6		14	15	16
00	01	02	03	04	05	0D	0E	0F

Figure 27. DDRAM spaces on LCD (Sitronix 2003).

When we want to use the specific place on the LCD to send data, we just select the position we want to deploy the number. For instance, if the position is “2”, we send the command (0x81) because DDRAM starts the positions from the 0x80 which is position “1”.

4.5. Modification Circuit

Necessary design of the expanded circuit needs to be done in order to have an accurate completed electronic equipment kit ready and adjustable. The program used to implement the adding circuits is Multisim 11.0v. It is a useful method for a designer as the functionalities are very easy to guide the user and set his/her own device. The most important part for a user to set a new circuit is to know exactly what is he/she going to design. To our case, a circuit which is compatible to the receiver part in order to view the incoming data to a display. From the transmitter part, a magnetic sensor to change the value of transmitting bits.

Accurate reading of the data sheets of both devices should be read in order to connect the node right with the rest materials. Data sheets could be found on Internet search engines by typing the exact code of the materials letter by letter. Drafts and schematics will take place in that part showing the connectivity of the MCU with the rest of the

implementing circuit. These schematics can be found from the data sheets of each part separately. A detailed design is shown in **Figure 28**.

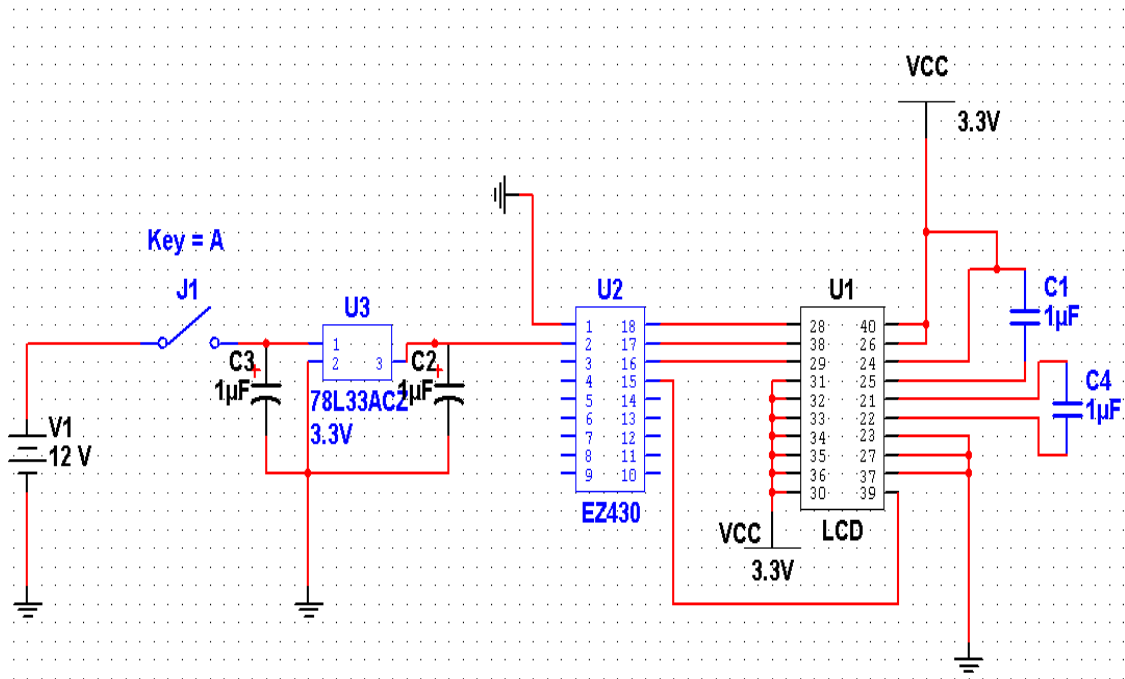


Figure 28. Receiver's node Layout.

This is the modified Receiver Node which allows the user to read the increasing value through the LCD.

4.6. Device Configuration

Every single device has its own MSP430F2274 MCU and CC2500 2.4GHz wireless transceiver chip (see **Figure 29**). The specific type of microcontroller is used for low power mode operations to save energy with efficiency. Easily adopted functionalities are detail explained in the chapter 5. Specific code blocks allow the general performance of the device to increase its durability. Various clock sources can be used for many clocking hardware blocks so that user application can select which one suits

the best requirements. Here the clocking system is used so efficiently that the minimum power is used to generate the PWM using the timer (Fletcher 1980: 630-643).

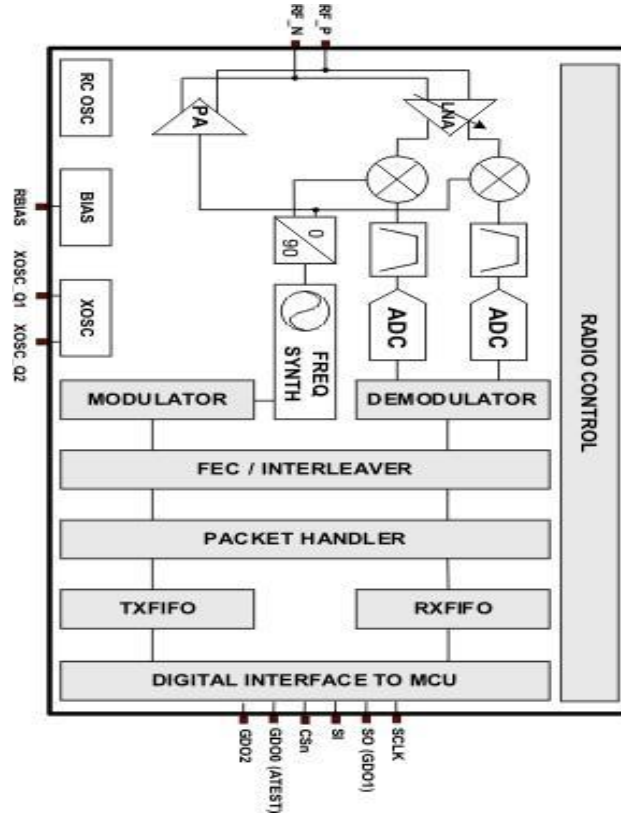


Figure 29. Internal chip CC2500 (Texas Instruments CC2500 2014).

The CC2500 comprises of a low intermediate frequency (IF) receiver. This received RF signal is directly amplified by the low noise amplifier (LNA). Next, comes the conversion of analog to digital signal so that it can be transmitted. Then, automatic gain control (AGC) action, channel filtering, demodulation, bit/packet synchronization are performed on digital form (Green 1995).

The flowchart in **Figure 30** describes what happens if the push button on the transmitter node is pressed.

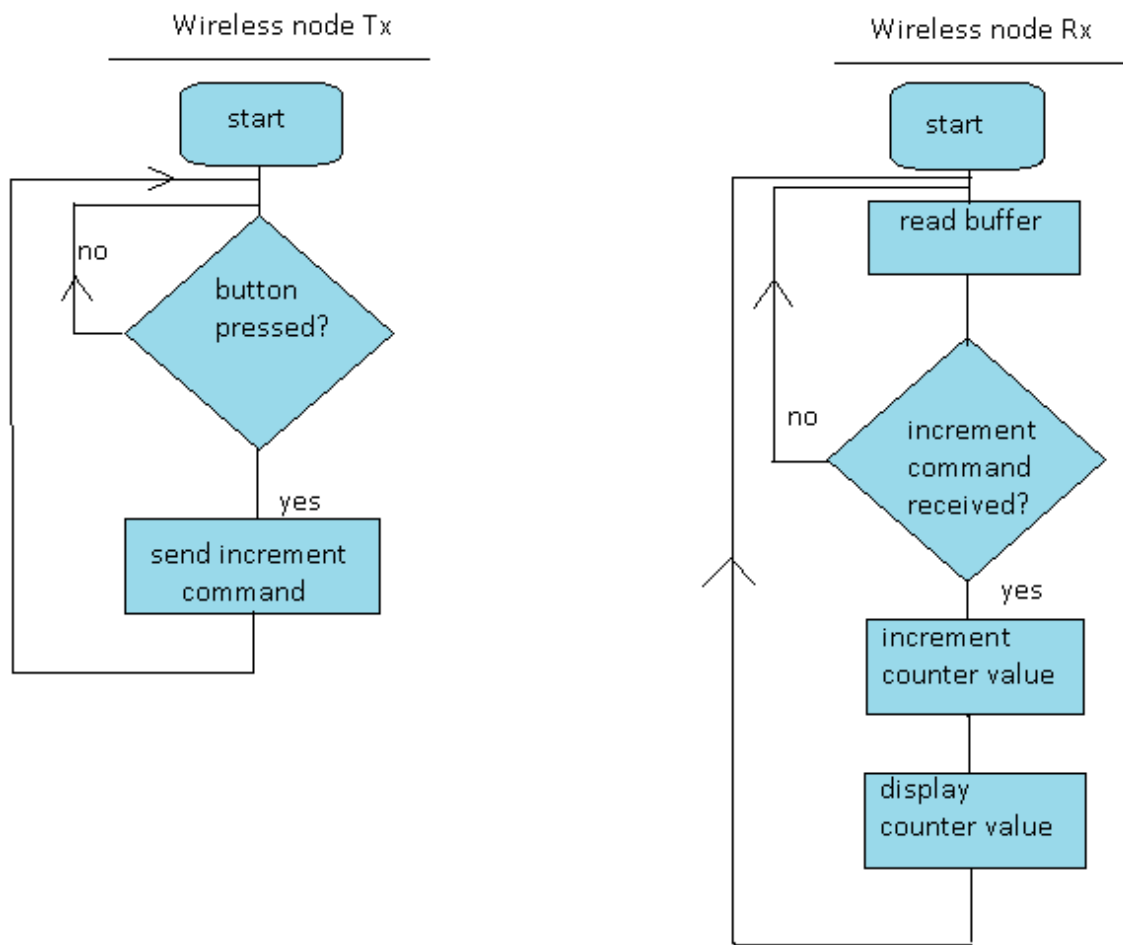


Figure 30. Flowchart for the program of the wireless nodes (transmitter and receiver).

4.7. Troubleshooting

The most challenging part in that procedure was to debug the program which corresponds to the initialization of the display and correctly handle the parity bits which should be enabled. Furthermore, because the user interface is Windows 7 all the programs, libraries and connections between the different infrastructures should perfectly match. For instance, Qt Creator needs a specific SQLite3 database so that it will open it through the main operations of the program. Moreover, when data comes in hexadecimal form, the specific type should be read from another compatible program. The user should find a way to combine the value again and form it in such way that will use it later for updating the server. For instance,


```
count_value = arrived.bytes(0)+arrived.bytes(1)
```

With that way the user accumulates the counter value in such way that it can be represented as a normal integer.

5. SOFTWARE

5.1. Meanings and Functions

Because of different program interfaces there may be slight changes in the main code blocks, as each kind of program has its own way to arrange functions, objects etc. PHP code blocks will be given as appendix in the end of that thesis. The codes in general follow a specific and simple to a user flowchart which describes each state of the program until the final outcome. The diagram in **Figure 31** shows how the MCU of a node handles the data (Yank 2004).

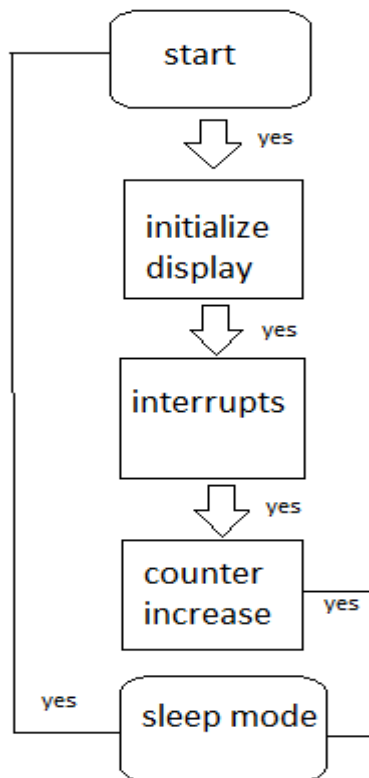


Figure 31. Flowchart (General Overview).

5.2. Initialization Process

The previous chapters gave a sum up of the necessary steps which must be followed in order to complete the microcontroller's functionalities. Code Composer Studio is a tool for the wireless node in order to compile the source code for the MCU of a wireless node. The unique style of coding in that software gives the potential to low current consumption for the device and easy way of implementing our functions inside the code.

First step is to create a main program which initialize the display and put it a state that we can read data on it. To make something like that, data sheets of the display EA DOGM163-A gives the reader all the necessary steps which should be activated in order to make it work as the user desires. Blinking cursor and backlight are just some functions that could be specifically written on the code so the display reacts. After the initialization functions of the display in our main program the counter should be created in order to count the data each time the RF MCU receives something. This is a closed loop counter but with a small specification. The counting data should be written in Hex code because it refers to the display.

This is the main body of our code to initialize the procedure of Radiofrequency communication of the node:

```

WDTCCTL = WDT PW + WDT HOLD; // Stop WDT
// 5ms delay to compensate for time to startup between MSP430 and
// CC1100/2500
__delay_cycles(5000);

TI_CC_SPISetup(); // Initialize SPI port
P2SEL = 0; // Sets P2.6 & P2.7 as
// GPIO
TI_CC_PowerupResetCCxxxx(); // Reset CCxxxx
writeRFSettings(); // Write RF settings to
// config reg
TI_CC_SPIWriteBurstReg(TI_CCxxx0_PATABLE, paTable,
paTableLen); // Write PATABLE
// Configure ports -- switch inputs, LEDs, GDO0 to RX packet info
// from CCxxxx
TI_CC_SW_PxREN = TI_CC_SW1; // Enable Pull up resistor
TI_CC_SW_PxOUT = TI_CC_SW1; // Enable pull up resistor
TI_CC_SW_PxIES = TI_CC_SW1; // Int on falling edge

```

```

TI_CC_SW_PxIFG &= ~(TI_CC_SW1);           // Clr flags
TI_CC_SW_PxIE = TI_CC_SW1;                 // Activate interrupt
                                           // enables
TI_CC_LED_PxOUT &= ~(TI_CC_LED1 + TI_CC_LED2); // Outputs = 0
TI_CC_LED_PxDIR |= TI_CC_LED1 + TI_CC_LED2; // LED Direction to
                                           // Outputs
TI_CC_GDO0_PxIES |= TI_CC_GDO0_PIN;        // Int on falling edge (end
                                           // of pkt)
TI_CC_GDO0_PxIFG &= ~TI_CC_GDO0_PIN;       // Clear flag
TI_CC_GDO0_PxIE |= TI_CC_GDO0_PIN;         // Enable int on end of
                                           // packet
TI_CC_SPIStrobe(TI_CCxxx0_SRX);            // Initialize CCxxxx in RX
                                           // mode.
                                           // When a pkt is received,
                                           // it will signal on GDO0
                                           // and wake up the
                                           // CPU

```

The next step is to initialize the UART function of the USB so that we can use the data for uploading them to the internet. To do that we should import the specific characteristics of the USB:

```

BCSCTL1 = CALBC1_1MHZ;                    // Set DCO
DCOCTL = CALDCO_1MHZ;
P3SEL |= 0x30;                             // P3.4,5 TXD/RXD
UCA0CTL1 |= UCSSEL_2;                      // SMCLK
UCA0BR0 = 104;                             // 1MHz 9600
UCA0BR1 = 0;                              // 1MHz 9600
UCA0MCTL = UCBSR0;                        // Modulation UCBSRx = 1
UCA0CTL1 &= ~UCSWRST;                     // Initialize USCI state

```

5.3. Interrupts and Counting Process

The previous chapter gave the specific idea behind the philosophy of the unique functionalities of the software. Under the right code, it can initialize our mechanisms and make them fulfill the device with the correct responds in order to generate our wireless counter device. Detailed code will be deployed and explained row by row for the needs of the thesis.

The code is ready to set up, initializing the functionalities of the pushing buttons. When a button is pressed, an interrupt comes indicating to the machine that incoming data came. From the USB part:

```

// The ISR assumes the interrupt came from a pressed button
#pragma vector=PORT1_VECTOR
__interrupt void Port1_ISR (void)
{
    // If Switch was pressed
    if(TI_CC_SW_PxIFG & TI_CC_SW1)
    {
        // Build packet
        txBuffer[0] = 3;                // Packet length
        txBuffer[1] = 0x01;             // Packet address
        txBuffer[2] = 0xFA;             // Load switch inputs
        txBuffer[3] = 0xD5;
        RFSendPacket(txBuffer, 4);      // Send value over RF
        __delay_cycles(5000);           // Switch debounce
    }
    txBuffer[2] = 0x00;
    txBuffer[3] = 0x00;
    TI_CC_SW_PxIFG &= ~(TI_CC_SW1);    // Clr flag that caused
                                        // interrupt
}

```

From the part of the End Device the code is:

```

#pragma vector=PORT2_VECTOR
__interrupt void Port2_ISR()
{
    char len;
    if(RFReceivePacket(txBuffer, &len))
    {
        if((txBuffer[1]==0xFA) && (txBuffer[2] == 0xD5))
        {
            lcd_update_function();
            while (!(IFG2&UCA0TXIFG));
            UCA0TXBUF = count_value >> 8;
            while (!(IFG2&UCA0TXIFG));
            UCA0TXBUF = count_value & 0xFF;
            __delay_cycles(500);
        }
        __delay_cycles(500);
        TI_CC_GDO0_PxIFG &= ((TI_CC_GDO0_PIN)<<1);
    }
}

```

For the End Device when magnetic sensor surface will attach to an iron surface then interrupt takes place (like pressing the push button) and the data through specific modulation transferred to the buffer of the Slave Device which is connected to the LCD and the number appears to the screen.

5.4. Database Connectivity

Putting things from the very beginning, the RF communication is working properly and the node sends and receives correctly the data. Second, an effective way to upload the counter value should be found in order to upload this number on a web page. To make something like that, we first need to access to the USB Serial Port of the Receiver part. Inside Qt Creator there is a .cpp file which handles the serial ports and detects a USB when it is connected on a PC. This is the function that enables reading data from the flash drive:

```
void serialhandler::openPort()
{
    usbport = new QSerialPort;
    // Example use QSerialPortInfo
    foreach (const QSerialPortInfo &info,
QSerialPortInfo::availablePorts()) {
        qDebug() << "Name      : " << info.portName();
        qDebug() << "Description : " << info.description();
        qDebug() << "Manufacturer: " << info.manufacturer();
        if(info.description()=="MSP430 Application UART")
        {
            usbport->setPort(info);
            if(!usbport->open(QIODevice::ReadWrite))
            {
                qDebug("Unable to open!");
            }
            else{
                qDebug("Opened the port");
                QSqlQuery query(mydb);
                while (query.next()) {
                    int count_value = query.value(1).toInt();
                    qDebug() <<count_value-1;
                }
            }
        }
    }
}
```

This function uses the serialhandler object to open the USB Ports putting some specific descriptions for the Node because it is the only one connected to the PC. Using the functionalities of Qt Creator the device can easily be detected through.

When the device is detected, then a message comes to the main window giving the required message to continue with the reading part. To read data, there should be a

“small” mechanism which enables Qt to modify them from hexadecimal to decimal. After the conversion, the counting value from the bale production is inserted into Qt Creator.

The following message is displayed when the “open device” button is pressed.

```
Database opened successfully
Name      : "COM4"
Description : "MSP430 Application UART"
Manufacturer: "Texas Instruments"
Opened the port
Name      : "COM3"
Description : "Intel(R) Active Management Technology - SOL"
Manufacturer: "Intel"
Name      : "COM1"
```

Next important factor is to reassure that this value that should always be up to date is saved and refreshed in our database. Using the source file that is created, database should communicate with Qt instantly every time the program runs. With that way:

```
void serialhandler::run()
{
    QSqlQuery query(mydb);
    while (1)
    {
        if(usbport->bytesAvailable())
        {
            arrivedBytes = usbport->readAll();
            count_value = arrivedBytes.at(0)*16 + arrivedBytes.at(1);
            qDebug() << count_value-1;
            c=count_value;
            c++;
            QString queryText = "UPDATE Bales SET c=" ;
            queryText.append(QString::number(count_value-1));
            qDebug() << queryText;
            query.exec(queryText);
        }
        QThread::msleep(20);
    }
}
```

Below it is depicted the outcome of the incoming data deployed in a sequence of counting procedure easily understandable from the user.

```
16  
"UPDATE Bales SET c=16"  
17  
"UPDATE Bales SET c=17"  
18  
"UPDATE Bales SET c=18"
```

With that function of the serialhandler, when USB has bytes in the buffer, Qt reads that value, converts it in a form readable from the user and updates the table with the contents of that value in the database. The database is up to date when incoming data is delivered. This is a simple user interface from Qt Creator able to interact with the user as shown in **Figure 32**. With a simple window the user activates the USB read-write function showing which USB is attached to the Port.

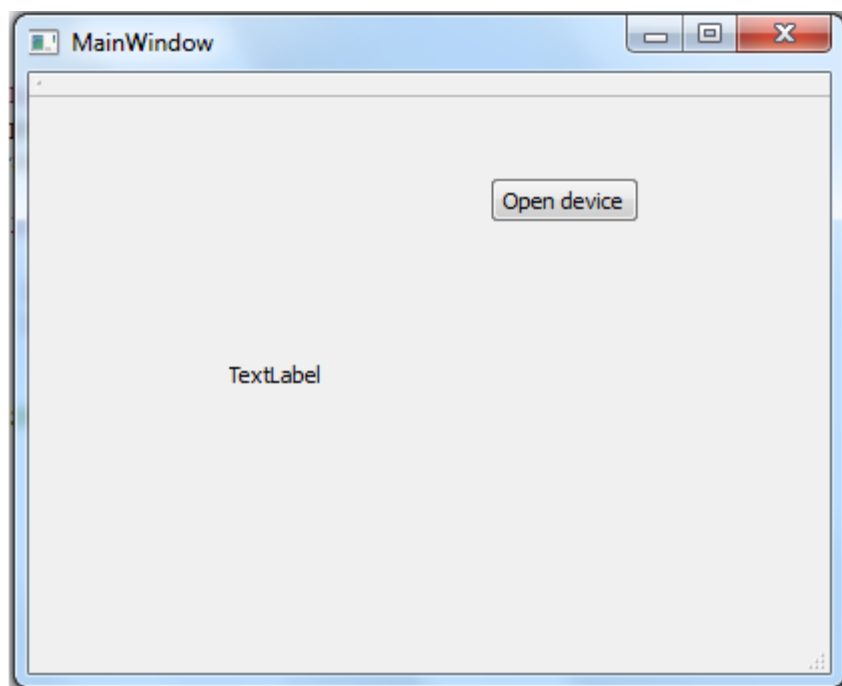


Figure 32. Main window which connects the sqlite3 database and the UART.

The potential for creating a most productive and illustrative interface is unlimited as the Qt Creator offers great functionalities for diagrams, clocks, counters, percentage bars etc.

5.5. Create SQLite3 Database on Windows

Web operations included servers, need to be activated. XAMPP application offers the cooperation of Apache, PHP and MySQL in such way that when a database should be updated, running of the XAMPP in first place. Provided that the connection is established the user can generate a database SQLite3 type as explained on previous chapters and open it through the Qt Creator by using the specific path.

5.6. Upload and Update the Database

When Qt Creator activates the UART with the wireless sensor, incoming data are on read state. By updating instantly every counting value, date and time the web interface can automatically be informed with these values as given in **Figure 33**. Now, the user can see the changes in the web browser by opening the page of the localhost provided by the XAMPP application. In the image below, the updated data take part in the inspection procedure of the crop production.

Act	Time	Bales
0	2013-12-10 13:22:08	4

Figure 33. Update values from PHP.

6. EXPERIMENTS AND RESULTS

6.1. Current Consumption

With the use of an electronic multimeter, the current values of the wireless node can be measured and used for comparing it with other wireless sensor nodes in the market.

MSP430F2274 offers different power configuration options for executing various functionalities according to the user specifications. It provides an active mode, sleep mode, and low power modes (LPM). In the LPM modes, only the necessary clock modules are running but **not** the CPU. In LPM the interrupts can be processed as they occur. **Table 1** summarizes the measured values for the current at an operating voltage of 3.3V and a frequency of 1MHz.

Table 1. Activation LPM current consumption

MCU MSP430F2274 mode	Stand by CC2500 mode	Current consumption in MSP430F2274
~1MHz	disabled	330 μ A
LPM1	disabled	295 μ A
LPM1	Yes	0.67 μ A
LPM4	Yes	0.35 μ A

The results show that the measured values are close to the values in the datasheet.

6.2. Validation Tests for the Reliability

The reliability of the transmission is very important. Wireless nodes with a high transmission loss are not suitable for the introduced application. **Table 2** shows the results of the reliability test for the wireless channel of the used wireless sensor nodes.

Table 2. Reliability of the wireless channel.

received packets/sent packets	received packets/sent packets	received packets/sent packets
57/60	60/60	59/60
55/60	55/60	60/60
58/60	57/60	54/60

The reliability can be computed with the following formula

$$Reliability = received\ packets / sent\ packets \quad (20)$$

From the above results it can be seen that the wireless transmission of the used nodes is quite high. Based on the reliability, the used wireless nodes are suitable for the introduced application.

6.3. Validation Tests for the Delay

During the transmission of data, some delayed outcomes occurred while the transmitting node communicates the receiver part. **Table 3** presents the amount of delayed packets out of 60 packets that have been sent.

Table 3. Lost data during transmission.

Tries	Amount of delayed packets out of 60
1 st	7
2 nd	11
3 rd	8
4 th	6
5 th	8
6 th	4

The delay of the packets can be caused by the buffer delay. However this affect has no serious influence for the introduced application because the delay time was minimal.

7. CONCLUSION AND FUTURE WORK

From year to year the crop production is increasing. In order to make the production process more efficient, a device, acting as a bale counter, has been implemented. The requirements for this device were low production costs, an easy installation and good usability. In addition the device should have the ability of saving data (amount of produced bales) to a database that can be accessed from almost everywhere in the world.

One aim of this thesis was to program the MCU of the eZ430-RF2500 wireless nodes to enable the communication between transmitter and receiver and to display the received data on the LCD screen. Another target was to write a software application that collects the data from the receiver node and writes it to a database. To make the data accessible on the World Wide Web a website has been developed that reads the data from the database. Furthermore the electronic required for the LCD screen has been designed.

The results in the experimental part show that the current consumption of the used wireless nodes is quite low and could be operated also with normal AAA size batteries instead of using the battery of the tractor. Furthermore the experiments of the wireless communication have shown good results.

As a future work the device can be optimized by upgrading wireless nodes with different kind of sensors according to the needs of the desired application. The implemented software for collecting and storing the data which is currently running on a PC can be ported to an Embedded Linux device such as raspberry PI. Also a 3G modem can be attached to an Embedded Linux device so that the data can be transferred directly from the monitoring device in the tractor to a server.

REFERENCES

- Bowick, Chris (1997). *RF Circuit Design*. Butterworth – Heinemann. Reprinted Edition. ISBN:0-7506-9946-9.
- Davis, Michele E. (2006) . *Learning PHP and MySQL* Sebastapol. ISBN: 0-596- 10110- 4.
- Electronic Assembly (2102). Dog series 3.3V incl. controller ST7036 FOR 4-/8bit spi (4-wire).
- Fletcher, William I. (1980). *An Engineering Approach to Digital Design*. Prentice-Hall N.J. ISBN:0-13- 277699-5.
- Floyd, Thomas L. (2002). *Electronic Devices*. 5th Ed. New Jersey :Persentage Hall. ISBN:0-13-973769-3.
- Google (2012). Method and apparatus for controlling a tractor/baler combination [online]. Available from the Internet : <URL: <http://www.google.com/patents/US6546705>>.
- Green, D.C. (1995). *Radio Systems of Technicians*. 2 Ed. Harlow :Longman. ISBN:0-582-24516-8.
- Greenfield, Josheph P. (1983). *Practical Digital Design Using ICs*. 2nd Ed. New York Press . ISBN-0-471-05791-6
- Gustafsson, Fredrik & Ljung Millnert (2010). *Signal Processing*. 2nd Ed. Malmo (Studentlitteratur). ISBN: 987-91-44-05835-1.
- Harikrishnan, Aswathanayaha T. & K.M Thayyib Sahini (2010). *Green Energy*. CRC Press .ISBN -13: 978-0-415-87628-5
- Honeywell (2013) .SS360PT/SS460P . High Sensitivity Latching Digital Hall-Effect Sensor ICs with Built-in Pull-up Resistor.
- Keneth, Kaiser L. (2004). *Electromagnetic Compatibility Handbook*. CRC Press Boca Raton . ISBN: 0-8493-2087-9.
- Lathi, B.P. (1998). *Modern Digital and Analog Communication Systems*. Oxford University Press. 3rd Ed. ISBN:0-19-511009-9.
- Mc Graw, Sabine Solomon (1999). *Sensors Handbook* . New York Hill. ISBN:0-07-059630-1.

- MegaSquirt (2013). Megasquirt and Microsquirt - Tach (RPM) input [online]. Available from the Internet: <URL: <http://www.msextra.com/doc/general/tachin.html>>.
- Microsoft Sync Framework Developer Center (2014). Introduction to Sync Framework Database Synchronization [online]. Available from the Internet: <URL:<http://msdn.microsoft.com/en-us/sync/bb887608.aspx>>.
- Microwaves &rf (2014).Building Wireless Sensor Networks [online]. Available from the Internet: <URL:<http://mwrf.com/systems/building-wireless-sensor-networks>>.
- Meloni, Julie,C. (2012). *PHP ,MySQL and Apache all in one*. 5th Ed. ISBN:987-0-672-33543-3.
- Messerle, Hugo K. (1994). *Magnetohydrodynamic Electrical Power Generation*. New York Press.ISBN: 0-471-94259-9.
- Mouser Electronics (2014). Board Mount Hall Effect/Magnetic Sensors [online]. Available from the Internet: <URL: <http://fi.mouser.com/ProductDetail/Honeywell/SS460P/?qs=sGAEpiMZZMs0JOhy9PM0UdwLAUVopG5O1xG7jdPAJX4%3d>>.
- Prodeveloper (2014). Open-Source Web Dev Stack in i5/OS [online]. Available from the Internet: <URL:<http://iprodeveloper.com/rpg-programming/open-source-web-dev-stack-i5os>>.
- Ranch Ramblins (2007). Hal's Thoughts On Country Living, Digital Photography, and Places of Interest [online]. Available from the Internet: <URL: <http://www.mitzenmacher.net/blog/?p=949>>.
- Sitronix (2003). ST7036 Dot Matrix LCD Controller/Driver V1.1 1.
- Stallings, William (2007). *Data and Computer Communications*. 8th Ed. New Jersey: Pearson Prentice Hall. ISBN: 0-13-243310-9.
- Stucky, Mathew (2002). *MySQL Building User Interfaces*. New Riders Publishing Indianapolis. ISBN:0-7357-1049-X.
- Texas Instruments (2007). eZ430-RF2500 Development Tool.SLAU227A.
- Texas Instruments (2007). Clear Channel Assessment Using TI MSP430 and CC2500 Radio. SLAA365.

- Texas Instruments Incorporated (2009). *eZ430-RF2500 Development Tool User's Guide* [online]. Available from the Internet: <URL: <http://www.ti.com/general/docs/lit/getliterature.tsp?baseLiteratureNumber=slau227&fileType=pdf>>.
- Texas Instruments Incorporated (2009). *eZ430-RF2500 Development Tool User's Guide* [online] Available from the Internet: <URL: <http://www.ti.com/product/cc2500>>.
- Texas Instruments Incorporated (2009). *eZ430-RF2500 Development Tool User's Guide* [online] Available from the Internet: <URL:<http://www.ti.com/lit/ds/symlink/cc2500.pdf>>.
- Welling, Luke & Laura Thomson (2008). *PHP and MySQL Web Development*. 4rth Ed. Indianapolis. ISBN: 0-672-32916-6.
- Yank, Kevin (2004). *PHP Database Driven Website using MySQL and PHP*. 3rth Ed. Richmond Sitepoint. ISBN:0-975-2402-1-8.

APPENDICES

APPENDIX 1. Initialization funtions for UART.

```

#include "msp430x22x4.h"

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;           // Stop WDT
    BCSCCTL1 = CALBC1_1MHZ;             // Set DCO
    DCOCTL = CALDCO_1MHZ;
    P3SEL = 0x30;                       // P3.4,5 = USCI_A0
                                         TXD/RXD

    UCA0CTL1 |= UCSSEL_2;               // SMCLK
    UCA0BR0 = 104;                      // 1MHz 9600
    UCA0BR1 = 0;                       // 1MHz 9600
    UCA0MCTL = UCBRS0;                 // Modulation UCBRSx = 1
    UCA0CTL1 &= ~UCSWRST;              // **Initialize USCI state
                                         machine**

    IE2 |= UCA0RXIE;                   // Enable USCI_A0 RX
                                         interrupt
    __bis_SR_register(LPM0_bits + GIE); // Enter LPM0, interrupts
                                         enabled
}

// Echo back RXed character, confirm TX buffer is ready first
#pragma vector=USCIAB0RX_VECTOR
__interrupt void USCI0RX_ISR(void)
{
    while (!(IFG2&UCA0TXIFG));          // USCI_A0 TX buffer
    UCA0TXBUF = UCA0RXBUF;              // TX -> RXed character
}

```

APPENDIX 2. CC2500 preferences modes.

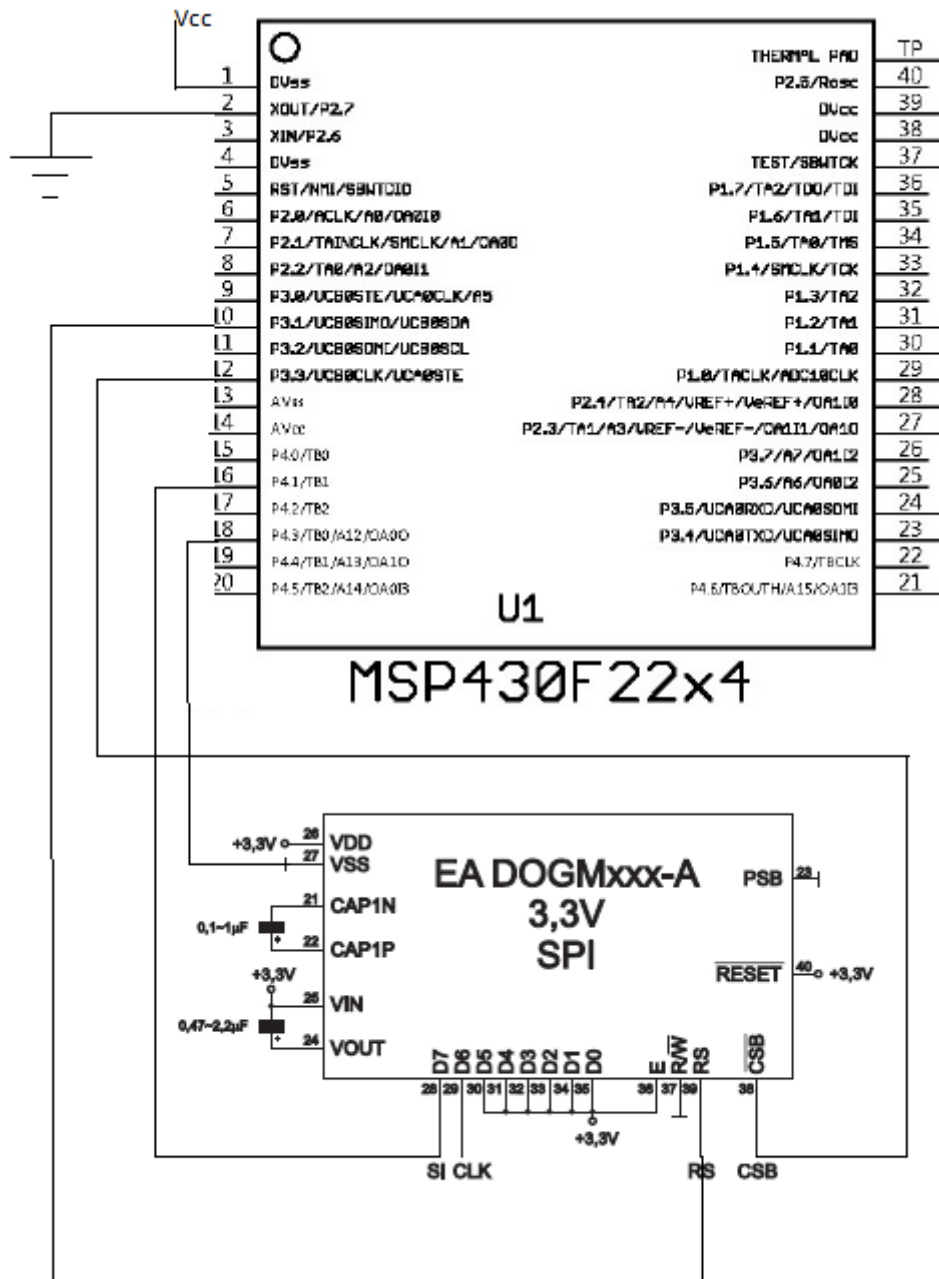
PARAMETER	MIN	TYP	MAX	UNIT	CONDITION
OPERATING CONDITIONS					
Operating supply voltage	1.8		3.6	V	
CURRENT CONSUMPTION					
RX input signal at the sensitivity limit, 250 kbps		16.6		mA	Optimized current
		18.8		mA	Optimized sensitivity
RX input signal 30 dB above the sensitivity limit, 250 kbps		13.3		mA	Optimized current
		15.7		mA	Optimized sensitivity
Current consumption TX (0 dBm)		21.2		mA	
Current consumption TX (-12 dBm)		11.1		mA	
RF CHARACTERISTICS					
Frequency range	2400		2483.5	MHz	
Data rate (programmable)	1.2		500	kbps	
Output power (programmable)	-30		0	dBm	
Sensitivity, 10 kbps		-99		dBm	Optimized current, 2-FSK, 230-kHz RX filter bandwidth, 1% PER
		-101		dBm	Optimized sensitivity
Sensitivity, 250 kbps		-87		dBm	Optimized current, 500-kHz RX filter bandwidth, 1% PER
		-89		dBm	Optimized sensitivity

APPENDIX 3.Pinout board.

Pin	Function	Description
1	GND	Ground reference
2	VCC	Supply voltage
3	P2.0 / ACLK / A0 / OA0I0	General-purpose digital I/O pin / ACLK output / ADC10, analog input A0
4	P2.1 / TAINCLK / SMCLK / A1 / OA0O	General-purpose digital I/O pin / ADC10, analog input A1 Timer_A, clock signal at INCLK, SMCLK signal output
5	P2.2 / TA 0 / A2 / OA0I1	General-purpose digital I/O pin / ADC10, analog input A2 Timer_A, capture: CC10B input/BSL receive, compare: OUT0 output
6	P2.3 / TA 1 / A3 / VREF - / VeREF - / OA1I1 / OA1O	General-purpose digital I/O pin / Timer_A, capture: CC11B input, compare: OUT1 output / ADC10, analog input A3 / negative reference voltage output/input
7	P2.4 / TA 2 / A4 / VREF + / VeREF + / OA1I0	General-purpose digital I/O pin / Timer_A, compare: OUT2 output / ADC10, analog input A4 / positive reference voltage output/input
8	P4.3 / TB0 / A12 / OA0O	General-purpose digital I/O pin / ADC10 analog input A12 / Timer_B, capture: CC10B input, compare: OUT0 output
9	P4.4 / TB1 / A13 / OA1O	General-purpose digital I/O pin / ADC10 analog input A13 / Timer_B, capture: CC11B input, compare: OUT1 output
10	P4.5 / TB2 / A14 / OA0I3	General-purpose digital I/O pin / ADC10 analog input A14 / Timer_B, compare: OUT2 output
11	P4.6 / TBOUTH / A15 / OA1I3	General-purpose digital I/O pin / ADC10 analog input A15 / Timer_B, switch all TB0 to TB3 outputs to high impedance
12	GND	Ground reference
13	P2.6 / XIN (GDO0)	General-purpose digital I/O pin / Input terminal of crystal oscillator
14	P2.7 / XOUT (GDO2)	General-purpose digital I/O pin / Output terminal of crystal oscillator
15	P3.2 / UCB 0SOMI / UCB 0SCL	General-purpose digital I/O pin USCL_B0 slave out/master in in SPI mode, SCL I2C clock in I2C mode
16	P3.3 / UCB 0CLK / UCA 0STE	General-purpose digital I/O pin USCL_B0 clock input/output / USCL_A0 slave transmit enable
17	P3.0 / UCB 0STE / UCA 0CLK / A5	General-purpose digital I/O pin / USCL_B0 slave transmit enable / USCL_A0 clock input/output / ADC10, analog input A5
18	P3.1 / UCB 0SIMO / UCB 0SDA	General-purpose digital I/O pin / USCL_B0 slave in/master out in SPI mode, SDA I2C data in I2C mode

Pin	Function	Description
1	P3.4 / UCA 0TXD / UCA 0SIMO	General-purpose digital I/O pin / USCL_A0 transmit data output in UART mode (UART communication from 2274 to PC), slave in/master out in SPI mode
2	GND	Ground Reference
3	#RST/SBWTIO	Reset or nonmaskable interrupt input. Spy-Bi-Wire test data input/output during programming and test
4	TEST/SBWTCK	Selects test mode for JTAG pins on Port1. The device protection fuse is connected to TEST. Spy-Bi-Wire test clock input during programming and test
5	VCC (3.6V)	Supply Voltage
6	P3.5 / UCA 0RXD / UCA 0SOMI	General-purpose digital I/O pin / USCL_A0 receive data input in UART mode (UART communication from 2274 to PC), slave out/master in in SPI mode

APPENDIX 4. SPI connection between MCU and LCD .



APPENDIX 5. PHP code for updating data and create a simple web page.

```

<?php
try
{
    //open the database
    $db = new PDO('sqlite:C:/sqlite/Newdata.db');
    //create the database
    $db->exec("CREATE TABLE Bales (a INTEGER PRIMARY KEY AUTOINCREMENT, b
datetime('now','localtime') , c queryText)");

    //insert some data...
    $db->exec("UPDATE Bales (Act,
Time, Bales) VALUES ('a', 'b', 'c')");
    //output the data to a simple html table...
    print "<table border=1>";
    print "<tr><td>Act</td><td>Time</td><td>Bales</td></tr>";
    $result = $db->query('SELECT * FROM Bales');
    foreach($result as $row)
    {
        print "<tr><td>".$row['a']. "</td>";
        print "<td>".$row['b']. "</td>";
        print "<td>".$row['c']. "</td></tr>";
    }
    print "</table>";
    // close the database connection
    $db = NULL;
}
catch(PDOException $e)
{
    print 'Exception : '.$e->getMessage();
}
?>

```

APPENDIX 6. General characteristics of CC2500.

Parameter	Min	Typ	Max	Unit
Frequency range	2400		2483.5	MHz
Data rate	1.2		500	kBaud
	1.2		250	kBaud
	26		500	kBaud

APPENDIX 7. Magnetic sensor and battery board footprints.

